

Sketch-to-Solution: An Exploration of Viscous CFD with Automatic Grids

Bil Kleb,^{*} Michael A. Park,[†] William A. Wood,[‡] Karen L. Bibb,[§] and Kyle B. Thompson[¶]
NASA Langley Research Center, Hampton, VA 23681, USA

Reynaldo J. Gomez III^{||}
NASA Johnson Space Center, Houston, TX 77058, USA

Steffan H. Tesch^{**}
Arizona State University, Tempe, AZ 85287, USA

Numerical simulation of the Reynolds-averaged Navier–Stokes (RANS) equations has become a critical tool for the design of aerospace vehicles. However, the issues that affect the grid convergence of three dimensional RANS solutions are not completely understood, as documented in the AIAA Drag Prediction Workshop series. Grid adaption methods have the potential for increasing the automation and discretization error control of RANS solutions to impact the aerospace design and certification process. The realization of the CFD Vision 2030 Study includes automated management of errors and uncertainties of physics-based, predictive modeling that can set the stage for ensuring a vehicle is in compliance with a regulation or specification by analysis). For example, the Cart3D inviscid analysis package has automated Cartesian cut-cell gridding with output-based error control. Fueled by recent advances in the fields of anisotropic grid adaptation, error estimation, and geometry modeling, a similar work flow is explored for viscous CFD simulations; where a CFD application engineer provides geometry, boundary conditions, and flow parameters, and the sketch-to-solution process yields a CFD simulation through automatic, error-based, grid adaptation.

I. Introduction

NUMERICAL simulation of the Reynolds-averaged Navier–Stokes (RANS) equations has become the modus operandi for the design and analysis of aerospace vehicles. For example, the aerodynamic database for the NASA Space Launch System (SLS) ascent aerodynamics and booster-separation [1] is built on thousands of RANS solutions with a significant investment in human and computer time. Despite its success, viscous Computational Fluid Dynamics (CFD) has not reached the level of automation desired by practitioners, primarily due to the lack of robust, efficient grid generation and error-based grid adaptation. Alauzet and Loseille [2] document the dramatic progress made in the last decade for solution-adaptive methods that includes the grid anisotropy to resolve simulations with shocks and boundary layers; and an informal Unstructured Grid Adaptation Working Group (UGAWG) has been formed to mature grid adaptation technology as described in their first benchmark paper [3]. A number of independent grid adaptation implementations (including the tools used in this work) have been verified for analytical solutions and shown to produce accurate results for a simple wing configuration [4]. The most recent UGAWG paper [5] details the complexities encountered in supporting realistic geometry encountered in typical use cases, namely Mechanical Computer-Aided Design (MCAD) integration. Taylor and Haines [6] provide an overview of geometry modeling concepts and requirements for computational simulation, motivated by the 1st AIAA Geometry and Mesh Generation Workshop (GMGW-1) [7]; and Park et al. [8] documented the state of solution-based anisotropic grid adaptation and

^{*}NASA Senior Researcher for Computational Aerothermodynamics, AIAA Associate Fellow.

[†]Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.

[‡]Assistant Branch Head, Aerothermodynamics Branch, AIAA Associate Fellow.

[§]Research Scientist, Aerothermodynamics Branch, AIAA Associate Fellow.

[¶]Research Scientist, Aerothermodynamics Branch, AIAA Member.

^{||} Aerodynamics Technical Discipline Lead, Aeroscience and Flight Mechanics Division, Senior AIAA Member.

^{**}Undergraduate student, Department of Mechanical Engineering.

motivated further development with the impacts that the improved capability would have on aerospace analysis in the broader context of the CFD Vision 2030 Study by Slotnick et al. [9].

The CFD Vision 2030 Study [9] states,

[U]ltimately the mesh generation process should be invisible to the CFD user or engineer. Given a suitable geometry representation and a desired level of solution accuracy, a fully automated meshing capability would construct a suitable mesh and adaptively refine this mesh throughout the solution process with minimal user intervention until the final accuracy levels are met. This process allows the user to focus on the final solution without concern for the construction and maintenance of the underlying mesh.

But as science fiction writer William Gibson says, “The future is already here—it’s just not very evenly distributed.” [10] For example, consider the inviscid flow simulation capability of Cart3D, which has automated, discretization-error-based grid adaptation [11]; the mostly-automated, mild-compressibility viscous capability of Lattice-Boltzmann flow solvers [12–14]; or the highly-automated viscous simulation capability of Dawes et al. [15, 16]. Each addresses one or more elements of the CFD Vision 2030 toward making the CFD grid “invisible;” but to date, none have all elements to achieve fully-automated viscous CFD, which is essential for crossing the chasm from an artisan product to a commodity solution.

By removing the human from the grid generation process, time-to-solution can be significantly reduced (months-weeks-days to hours) and a standardized, repeatable workflow process can be established (and optimized). This paper seeks to explore work flows for geometry-to-viscous CFD solution, without an explicit human-in-the-loop grid generation step, via grid adaptation driven by the L_2 norm of solution interpolation error. Or as the title states, “automatic grids” where automatic is used in the sense of, “a device or process working by itself with little or no direct human control.” [17] Toward this end, we will apply sketch-to-solution work flows to a menagerie of geometries and flow regimes to uncover issues spanning initial grid generation, flow solver robustness, grid adaptation mechanics, geometry recovery, and error estimation. From this, we will then try to isolate causes of failure, identify areas for investment, and seek workarounds to make near-term progress.

II. Method

The components of solution-based grid adaptation are shown in Figure 1. Starting with a problem definition, an initial grid is created and a flow solution is computed. The information from the flow solution is used to estimate error and specify a new grid resolution request. If the estimated errors are larger than limits specified by the practitioner, the current grid system is modified by grid mechanics to adhere to the grid resolution request.* The process is repeated until an exit criteria is met (e.g., accuracy requirement or resource limit).

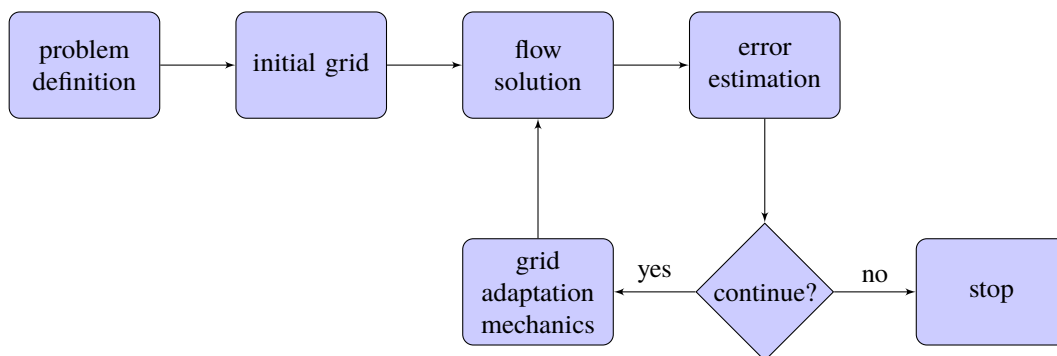


Fig. 1 Solution-based grid adaptation process.

The components of the particular sketch-to-solution work flows we are employing are described in the following subsections.

*Note that the grid mechanics could seek this goal by 1) altering the existing grid, 2) generating a new grid, or 3) combining (1) and (2). In this work, option (1) is employed, i.e., altering the existing grid.

A. Problem Definition

Engineering Sketch Pad (ESP) [18], which includes the Open-source Constructive Solid Modeler (OpenCSM) [19], is used for geometry creation and import. It has the ability to create geometry through sketching and boolean operations between primitive geometries such as boxes, spheres, cones, and so forth. It also has the ability to import STEP (Standard for the Exchange of Product model data) geometry definitions exported by other MCAD packages.

To define boundary conditions, ESP is used to assign attributes to geometry faces, which are subsequently parsed to create boundary specifications for the flow solver. And finally, the flow conditions are described in a format according to the flow solver employed.

B. Initial Grid

The initial grid is produced in two phases: 1) surface grid and 2) volume grid. The surface grid begins with the tessellation object given by Electronic Geometry Aircraft Design System (EGADS) [20], which is part of the ESP. This is largely used for visual rendering purposes and is not suitable for CFD. So, the grid mechanics described in a following subsection are used on this initial surface grid to satisfy geometry constraints like high curvature and unresolved features such as tiny edges or faces. (Further details are available in [5].) Next, the volume is filled with *isotropic* tetrahedra by AFLR3 [21] or TetGen [22]. The surface (with EGADS geometry associativity) and volume grids are then merged and fed to the grid mechanics to produce a grid that also satisfies similar geometry constraints that were imposed on the surface grid. (Again, for more detail, see [5].)

C. Flow Solution

Two flow solvers are used in this work. The first, FUN3D-FV [23, 24], is a finite-volume Navier-Stokes solver in which the flow variables are stored at the vertices of the grid. At interfaces between neighboring control volumes, the inviscid fluxes are computed using approximate Riemann solvers based on solution values from either side of the interface. For second-order accuracy, interface values are extrapolated from the vertices with gradients derived from an unweighted least-squares technique [23]. The full viscous fluxes are discretized using a finite-volume formulation in which velocity gradients on the dual faces are computed using the Green-Gauss theorem. The solution at each time step is updated with a backward Euler time-integration scheme where the linear system of equations is approximately solved with a multicolor point-implicit procedure [25]. The Spalart-Allmaras (SA) turbulence model [26] is loosely-coupled to the meanflow equations.

The SA turbulence model requires the distance from every vertex to the nearest no slip boundary condition. The standard wall distance calculation in FUN3D-FV finds the nearest surface vertex and then searches adjacent triangles to see if they are closer than the closest surface vertex. This standard wall distance method is inaccurate if the closest triangle is not adjacent to the closest surface vertex, which is frequently the case for highly-anisotropic adapted grids. So, to provide an accurate wall distance, which is critical to the SA model, an alternative method is used for adapted grids [27]. The alternative method encloses each surface triangle in a bounding box, which is stored in an Alternating Digital Tree (ADT) [28], to rapidly find the closest surface triangle for adapted unstructured grids.

The second flow solver used herein, HyperSolve, is an edge-based finite-volume Navier-Stokes solver. Both the inviscid and viscous fluxes are computed at the interfaces of dual control volumes around each vertex in the grid. Following the approach of Barth [29], the inviscid fluxes at interfaces of neighboring control volumes are computed by extrapolating the solution variables to the interface location. This extrapolation uses gradients computed by an unweighted least-squares technique. The viscous fluxes are computed at the same interface as the inviscid fluxes, using the alpha-damping technique by Nishikawa [30].

HyperSolve is purely a discretization library, and gains grid adaptation through the Dependency Inversion Principle (DIP) by being a plugin in T-infinity [31]. Likewise, HyperSolve also gains access to the PETSc library [32, 33] through the DIP by means of the Enigma library [34]. By using Enigma, HyperSolve can employ a variety of nonlinear solvers, including the Jacobian-Free Newton-Krylov technique and defect correction technique.

D. Error Estimation

Error estimation is accomplished by forming a Riemannian metric tensor to control the L_2 norm of the interpolation error of a solution scalar field. Classic Hessian-based adaptation methods [35, 36] control quadratic interpolation error estimates in the L_∞ -norm, which results in excessive refinement of nonsmooth regions. To form the metric, a Hessian is reconstructed by recursive application of the L_2 -projection gradient reconstruction scheme. The gradient is computed in each element and a volume-weighted average is collected at each vertex [37]. The second-derivative Hessian terms

are formed by computing the reconstructed gradients of these gradients formed in the first pass. The mixed derivative terms of the Hessian are averaged. A special boundary treatment is employed, where the reconstructed Hessian on the boundary is replaced with an extrapolation from neighboring interior vertices, which have a well-formed stencil.

The Hessian is then decomposed into eigenvalues and eigenvectors. The metric is formed by recombining the absolute value of the eigenvalues with the eigenvectors to ensure the metric is symmetric positive definite. The metric at each vertex is scaled to control the L_2 norm [37], creating what is termed a multiscale metric; and the gradation of the metric field is limited to 3.0 isotropically in the metric space [38]. The complexity, proportional to the total number of vertices in the adapted grid, is computed; and the metric is globally scaled to set its complexity to a requested value.

E. Grid Adaptation Mechanics

The *refine* grid adaptation mechanics package[†] is used herein. It is designed to output a unit grid [39] in a provided metric field from an initial grid that conforms to the geometry. Geometry is accessed by *refine* through the EGADS application programming interface.[‡]

The standard practice of reconstructing the grid association to the topology and parameters of the MCAD model is error prone. While *refine* has a rudimentary MCAD association recovery method, the information describing this association can be persisted to eliminate the possibility of MCAD association error. The Gamma Mesh Format is used for grid, geometry association, solution, and metric interchange.[§] *refine* can read and write this format natively to interact with a single file for the whole domain during parallel execution. *refine* maintains a list of geometric associations at each discrete vertex, where a vertex can have zero or more associations. These associations can have zero or more geometry nodes, edges, or faces. The associations record the type of parent geometry entity, an index to the instance of the entity, and the parameters of the entity. The topology exposed by EGADS is traversed to mark discrete vertices with parametric periodicity or degeneracy [5]. This gives *refine* awareness of parametric discontinuities and allows for correct evaluation (and inverse evaluation) of existing and new discrete vertices. This association is also used to prevent grid modifications that would violate the one-to-one connection of discrete grid and geometry model topology (e.g., removing or moving a vertex associated with a geometry model node).

refine uses the combination of split, collapse, and element swap operations [41]. Some of these classic operators are undergoing replacement with cavity operators [42]. Vertex relocation is performed to improve adjacent element shape with a convex combination of ideal vertex locations [43] or nonsmooth optimization based on Freitag and Ollivier-Gooch [44]. The geometry constraints introduced in the initial grid section remain active during all subsequent adapted grids [5].

III. Results

To produce results with the above work flow, the CFD application engineer defines the problem by creating the geometry in ESP (with the option to import geometry via STEP), specifying the boundary conditions, and creating the necessary configuration inputs for the flow solver and *refine*, and then starts the flow solver-*refine*-EGADS adaptation process. A typical workflow for FUN3D-FV might look like,

```
project=sandiacone

# Create EGADS geometry and notional FUN3D BC tags from OpenCSM file
csm2mapbc-v114.sh ${project}.csm > ${project}01.mapbc

# edit ${project}01.mapbc appropriately
# create FUN3D input file: fun3d.nml
# create FUN3D-refine file: case_specifics

# Create curvature-constrained surface mesh and initial TetGen volume mesh
ref bootstrap ${project}.egads

# Use refine to propagate curvature-constraints into the volume mesh
ref_driver -i ${project}-vol.meshb -g ${project}.egads -o ${project}01
```

[†]*refine* is developed by NASA and is available via <https://github.com/NASA/refine> under the Apache License, Version 2.0, last accessed April, 2019.

[‡]Or the EGADSLite [40] version of EGADS, which is a lightweight ANSI-C version of the EGADS functions required by the initial and adaptive grid process, where the target is any high-performance computing equipment currently in use, including graphic processing units (GPUs).

[§]Documentation and reference implementations of readers and writers are available at <https://github.com/LoicMarechal/libMeshb>, last accessed April, 2019.


```
# Start refine-FUN3D-EGADS adaptation process (f3d comes with FUN3D)
f3d start
```

A variety of cases have been simulated to explore the robustness, cost, and accuracy of adaptive grid mechanics coupled to geometry with ESP/EGADS, a flow solver, and *refine*. To date, the following configurations have been investigated:

- A Mars ascent vehicle with reaction control system (RCS) jets
- A parachute test vehicle
- A cone-cylinder-flare with jet in crossflow [45]
- A 65-degree sharp leading-edge delta wing [46]
- A hypersonic cylinder
- The benchmark supercritical wing [47]
- The Transonic Dynamics Tunnel [48]
- AEDC Tunnel D [49]
- ONERA M6 wing [50]
- The C25F and C25P supersonic aircraft configurations [51]
- The NASA Juncture Flow Model [52]
- The Acoustic Research Nozzle #2 [53]
- The Drag Prediction Workshop (DPW)-6 Common Research Model (CRM) [54]
- The High Lift Prediction Workshop (HLPW) HL-CRM(v2) configuration [55]
- The JAXA Standard High Lift configuration (with and w/o nacelle) [55]
- The Sonic Boom Prediction Workshop (SBPW3) C608 configuration
- The SBPW3 9×7 biconvex configuration [56]
- The sonic boom SEEB-ALR configuration [51]
- The AGARD-138 B4 wing body [57]
- A supersonic five-hole probe calibration database
- A supersonic retropropulsion configuration
- The X-38 [58]
- A smooth-body Orion heat shield
- The SLS with protuberances
- The Stardust sample return capsule
- An F-15 with shock sensing air-data probe [59]
- A Winnebago
- The NASA LOFTID configuration [60]
- The Langley Unitary Plan Wind Tunnel (LUPWT) [61]
- The LUPWT check standard model [62]
- The NASA HL-20 [63]
- The Oberkampf and Aeschliman sphere cone [64]
- A 70-degree sphere-cone capsule
- The OpenCSM lander [19]
- Mars InSight ground wind loads [65]
- The OpenCSM “dragon” capsule
- Several proprietary geometries

Of course not all have been fully successful for a variety of reasons. The most common failing stems from the quality of the geometry model. Many of the issues encountered, such as degenerate points, periodic faces, tiny edges, and sliver faces, have been addressed during this exploration phase and are documented in [5]. In summary, geometry models

created natively in ESP have consistently worked but those imported via a STEP file created by a foreign mechanical computer-aided design (MCAD) tool, are not as robust. To give the reader a feel for issues uncovered along the way and the current state of this workflow, six of the above explorations are examined in the following subsections.

A. Joint Computational/Experimental Aerodynamics Program Sliced Sphere-Cone with Flap

The first explorative case for the sketch-to-solution work flow is the 1992 Joint Computational/Experimental Aerodynamics Program (JCEAP) conducted at Sandia National Laboratories by Oberkampf and Aeschliman [64]. The wind tunnel model is a spherically-blunted 10° half-angle cone with a longitudinal slice, an axial length of 26.4 cm, and three detachable flaps with deflection angles of 10, 20, and 30 degrees. Tests were conducted in the Sandia National Laboratories Hypersonic Wind Tunnel at freestream condition $M_\infty = 7.84$, $\alpha = -10^\circ$ to $+18^\circ$, and $Re_L = 1.80 \times 10^6$. The test article wall temperature is reported by Walker [66] as 317 K. The present simulations assumed steady, laminar, perfect gas air. Force and moment quantities, including the normal force (C_N), axial force (C_A), and pitching moment (C_m) were measured, while the forebody axial force coefficient ($C_{A, \text{forebody}}$) and the nondimensional axial location of the center of pressure (X_{C_p}/L), were derived. For the JCEAP experiment, a base plate was set on the sting 1.651 mm aft of the test article, to isolate the base pressure. Pressure transducers measured the base pressure in the gap between the model and the base plate; and as a data reduction step, the base pressure was subtracted from the axial force coefficient, resulting in the “forebody axial force coefficient ($C_{A, \text{forebody}}$).” JCEAP used this $C_{A, \text{forebody}}$ for CFD validation because their computational approach did not include the model base. However, the present results do include the model base, but do not include the base plate nor the sting.

The investigation was performed using the sketch-to-solution workflow as illustrated in Figure 1 and described in Section II. The problem definition began with the sketch of the JCEAP geometry from Figure 1 of Ref. [64], which was turned into a fully-parameterized computational geometry with boundary conditions by a novice user of ESP within an afternoon (Figure 2). Appendix A contains the resulting OpenCSM geometry model file.

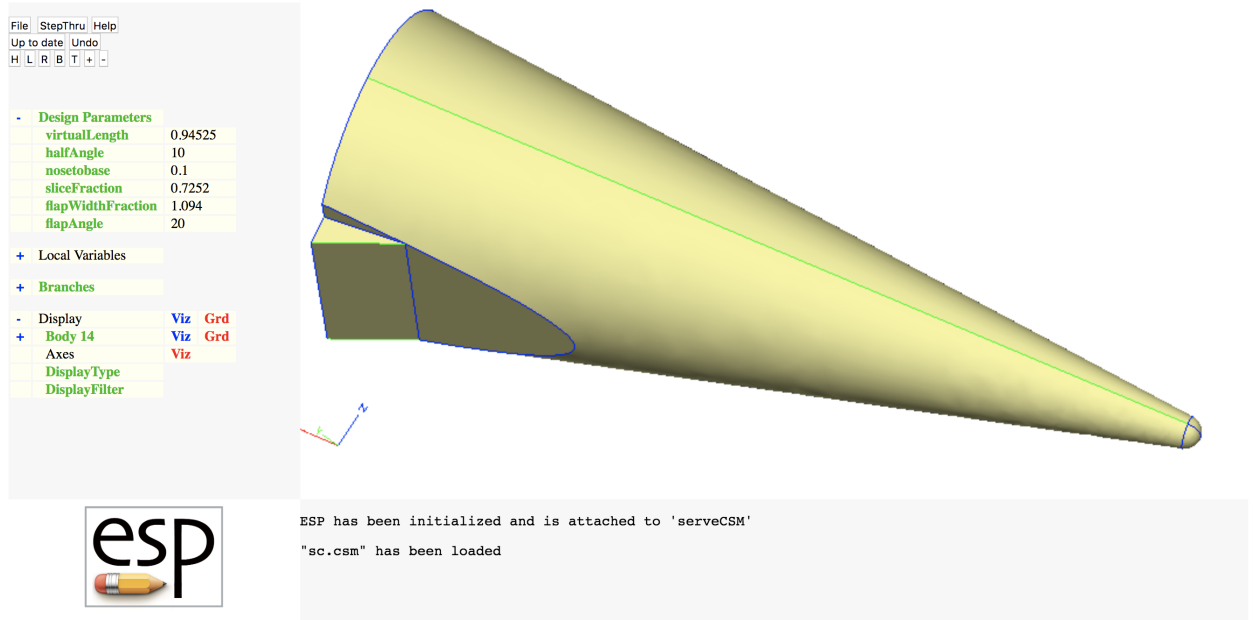


Fig. 2 Sketch-to-solution geometry for the sliced sphere-cone JCEAP geometry of Oberkampf and Aeschliman [64].

Four cases were simulated at 10° angle of attack, for each of the three flap attachments as well as flapless. The *refine* grid adaptation mechanics package completed 45-grid adaptation cycles for all cases, resulting in final grids of approximately 10M vertices.

Convergence of the solution-adaption process is plotted in Figure 3 for the flapless case, plotting computed aerodynamic coefficients versus number of grid vertices, relative to the JCEAP measurements. Figure 4 separates out each aerodynamic metric over the later stages of the solution-adaption process, for grids between 1M and 10M vertices. While $C_{A, \text{forebody}}$ and C_N are predicted to within experimental uncertainty, the pitching moment prediction is off by

6% from the experimental measurement, a disappointing amount. Table 1 lists the agreement between simulation and measurement of the aerodynamic coefficients for all flap deflection angles.

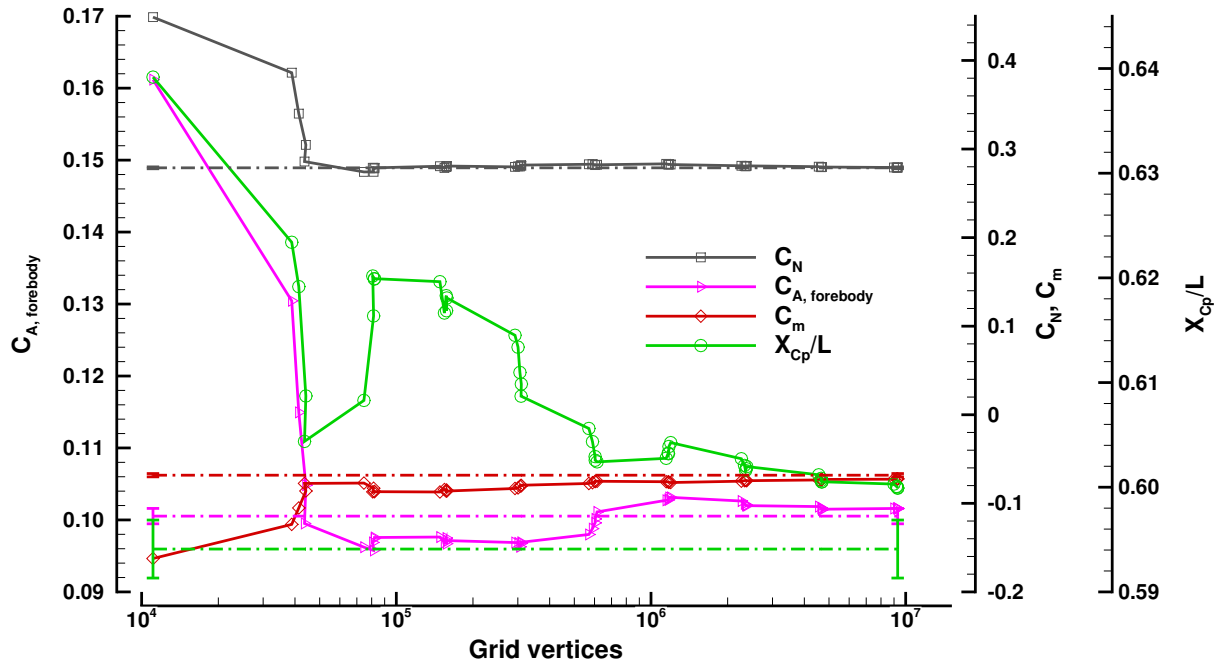
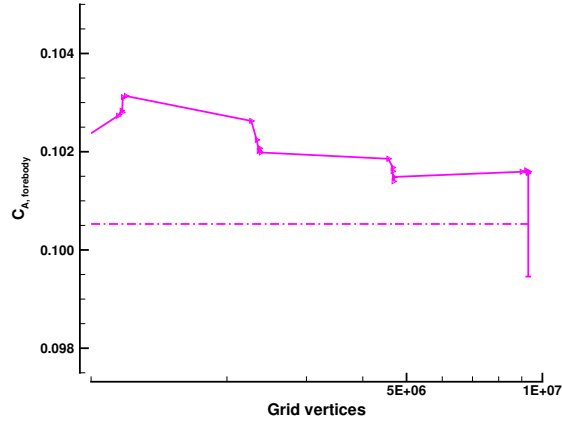
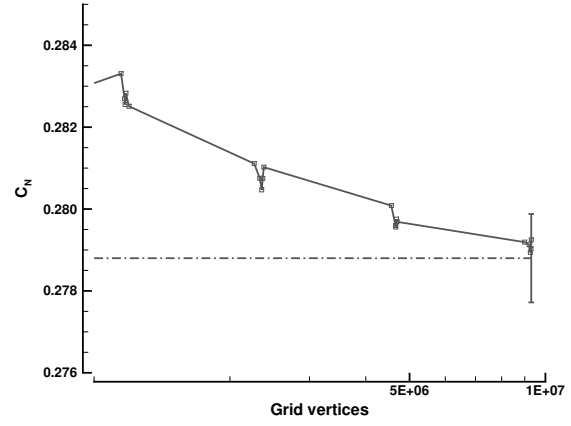


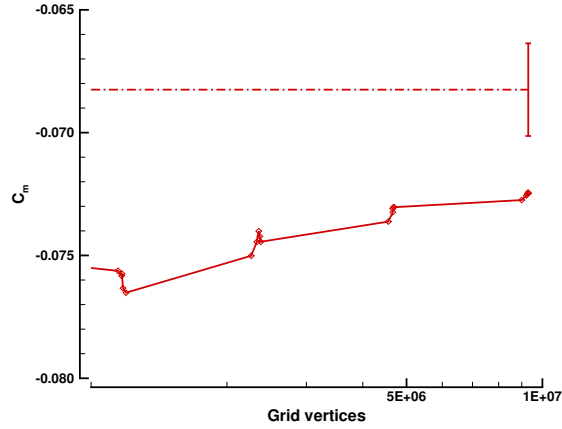
Fig. 3 Sketch-to-solution process convergence on JCEAP Mach 8, flapless case versus number of grid vertices. Experimental measurements shown as dash-dot lines with 2σ uncertainty bars from [64].



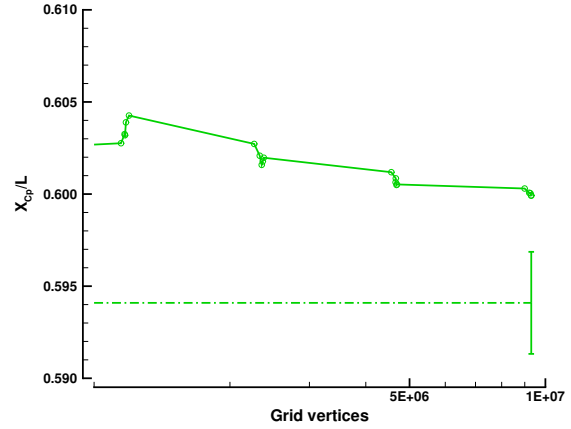
(a) Forebody Axial Force Coefficient.



(b) Normal Force Coefficient.



(c) Pitching Moment Coefficient.



(d) Nondimensional Center of Pressure Location.

Fig. 4 Process convergence from 1M to 10M vertices for JCEAP Mach 8, flapless case. Experimental measurement shown as dash-dot line with 2σ uncertainty bar from [64].

Table 1 Aerodynamics for JCEAP experiment at Mach 8, 2M Reynolds number per foot, 10-degrees angle of attack compared with Oberkampf and Aeschliman [64] subject to a 10M-vertex grid resource limit.

		Flap Deflection Angle			
		0°	10°	20°	30° [†]
$C_{A, \text{forebody}}$	Exp.	0.1005	0.1069	0.1224	0.1689
	CFD	0.1016	0.1074	0.1251	0.1821
	% error	1.1	0.4	2.2	7.9
C_N	Exp.	0.2788	0.3158	0.3561	0.4307
	CFD	0.2792	0.3065	0.3547	0.4493
	% error	0.2	-3.0	-0.4	4.3
C_m	Exp.	-0.0683	-0.1069	-0.1596	-0.2695
	CFD	-0.0725	-0.1064	-0.1662	-0.3014
	% error	6.2	-0.5	4.3	11.9
X_{C_p}/L	Exp.	0.5941	0.6313	0.6727	0.7417
	CFD	0.5999	0.6336	0.6804	0.7582
	% error	1.0	0.4	1.1	2.2

[†] The 30° flap deflection simulation was unsatisfactorily convergent on a 10M vertex grid, so the grid refinement process was continued until computational resource limits were hit with 45M vertices, still experiencing unsatisfactory process convergence. The results are included herein for completeness, but are not recommended.

B. Laminar Flow over Delta Wing

To explore the robustness and cost of the adaptation process, the adaptation of laminar flow over a delta wing case is studied for two typical solver techniques: defect correction and Jacobian-Free Newton-Krylov (JFNK). This case was originally described by Wang et al. [67] and subsequently modified by Klaij et al. [68], based on the experiment of Riley and Lowson [69]. The freestream conditions are Mach 0.3, 4000 Reynolds number based on the root chord length, and 12.5° angle of attack. Isothermal noslip boundary conditions are set to freestream temperature; the Prandtl number is 0.72; and the viscosity is assumed to be constant.

The HyperSolve flow solver was used with *refine* to adapt from an initial volume grid of 143 vertices. Figure 5 shows the surface grid of the delta wing geometry for the initial and 19th grid in the adaptation cycle. Figure 6 shows

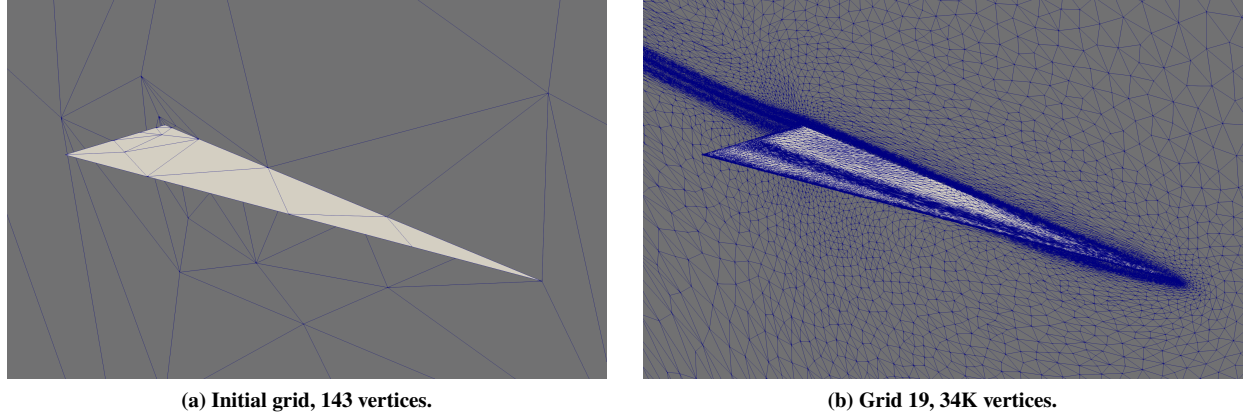
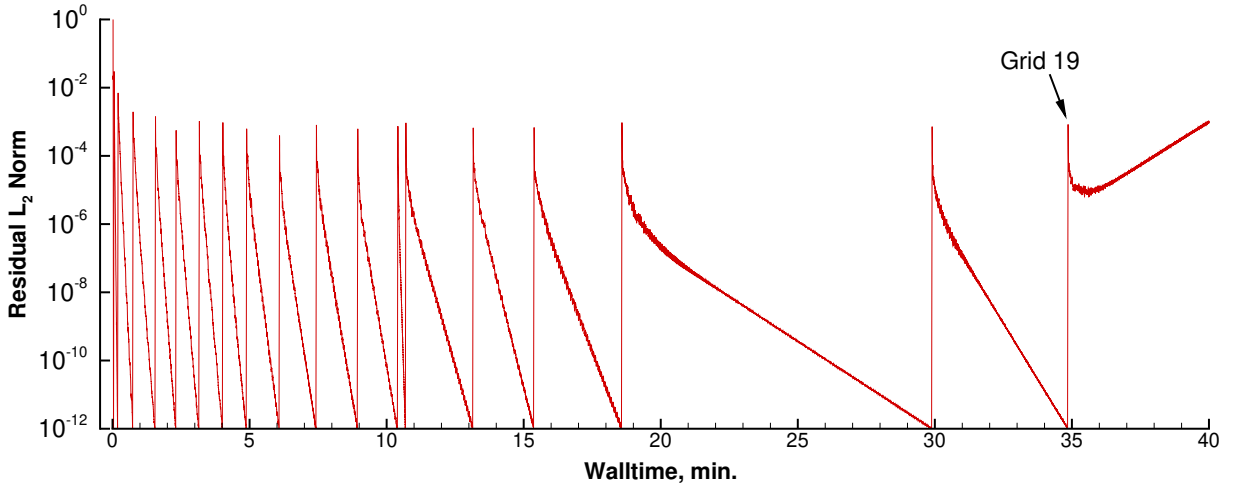
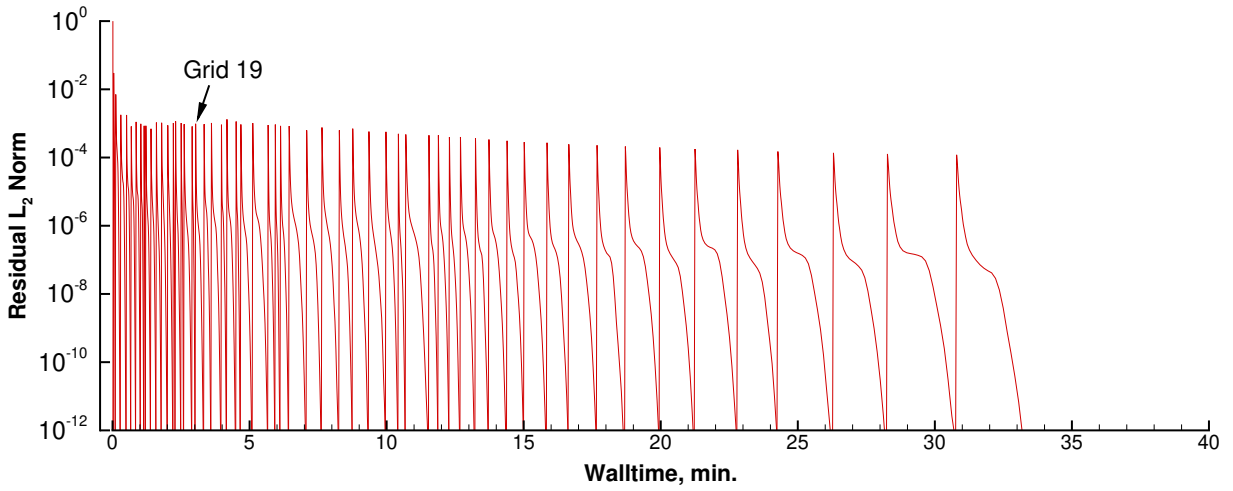


Fig. 5 Close view of delta wing surface grid.

the two fold advantage of using a JFNK solver over a defect-correction solver: 1) increased robustness and 2) increased efficiency. The implementation details of both solvers are detailed in Ref. [34]. In both solvers, the solution on the previous grid is interpolated onto the subsequent grid to provide a “warm restart.” This is particularly beneficial for the JFNK solver, because it effectively serves as a grid-sequencing technique for globalizing Newton’s Method, which significantly decreases the time spent before obtaining quadratic convergence. The quadratic convergence rate of the JFNK solver significantly outperforms the linear convergence rate of the defect correction solver, with the JFNK solver reaching the 19th adapted grid in less time than it takes the defect correction solver to reach the 6th grid. The solution interpolation is insufficient to ensure convergence of the defect correction solver, and by the nineteenth grid the defect correction solver diverges. This divergence manifests as a constant increase in the L_2 norm of the residual, eventually terminating the entire adaptation process due to the generation of negative densities in the solution.



(a) Defect Correction.



(b) Newton-Krylov.

Fig. 6 Comparison of the robustness and cost for two solver approaches.

C. AGARD Wing Body

An AGARD wind tunnel test [57] of an RAE wing with an axisymmetric body was chosen to examine the ability of the solution process to predict distributed pressures on a configuration at transonic conditions, namely case number 6, at a Mach number of 0.9, with angle of attack of 1° , $T_o \approx 320$ K and a Reynolds number of 1×10^6 based on the geometric mean chord of the gross wing. The geometry was modeled in ESP based on the sketch shown in Figure 7 and tabulated points on an RAE 'A' airfoil.

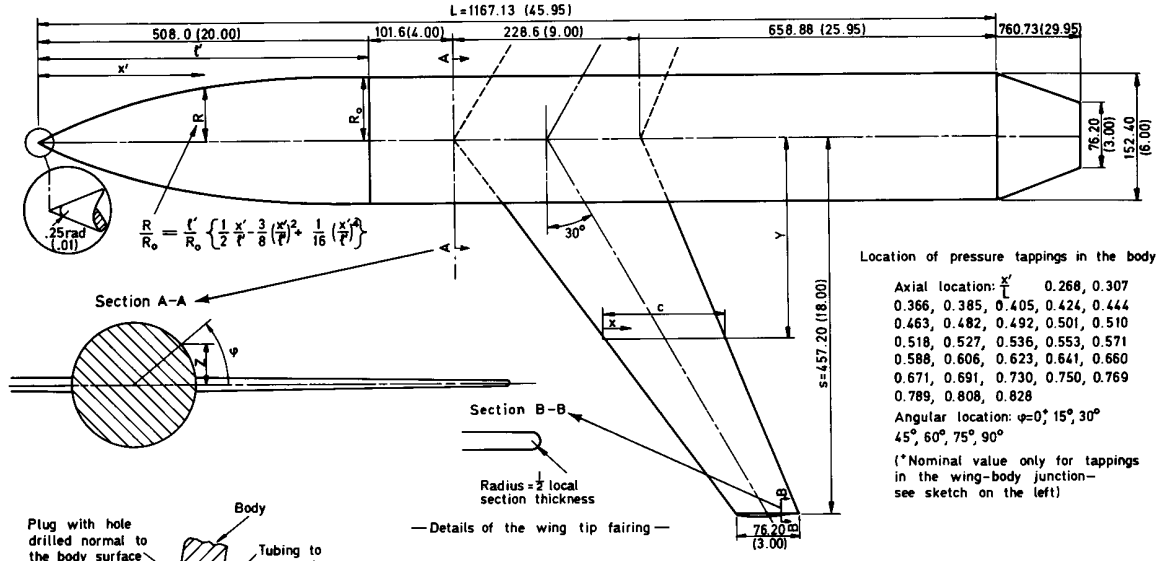


Fig. 7 AGARD wingbody planform drawing from [57].

Figure 8 shows the OpenCSM model in the ESP web browser viewer. A spherical farfield boundary was generated 40 body lengths away from the center of the body. The OpenCSM script is 215 lines long, including 93 lines for the wing profile, and the farfield boundary geometry definition—see Appendix B for listing. The initial volume grid for this solution contained 48K vertices and 277K tetrahedra, and the final grid contained 1.8M vertices and 9.9M tetrahedra after the 20th grid adaptation. Figure 9 shows the grid on the initial and final (adapted) grids.

Figure 10 shows the pressure distribution over the entire wingbody geometry and wing pressure distributions over the starboard wing, while Figure 11 depicts every 4th grid adaptation on the seven instrumented wing stations on this model, including the reported uncertainty in $C_p = \pm 0.01$. The results show monotonic convergence toward the wind tunnel reported pressure coefficient and the final results compare well to the reported experimental uncertainties.

To show that the FUN3D-FV+refine+EGADS process is recovering the shock boundary layer interaction effects, this case was rerun with an Euler code, Cart3D [70], using C_N as a functional to drive the adjoint grid adaptation process. The Cart3D process requires an order of magnitude less elapsed time and an order of magnitude fewer processors, but clearly does not capture the shock boundary layer interaction correctly due to the lack of viscous terms in the solver as seen in Figure 12.

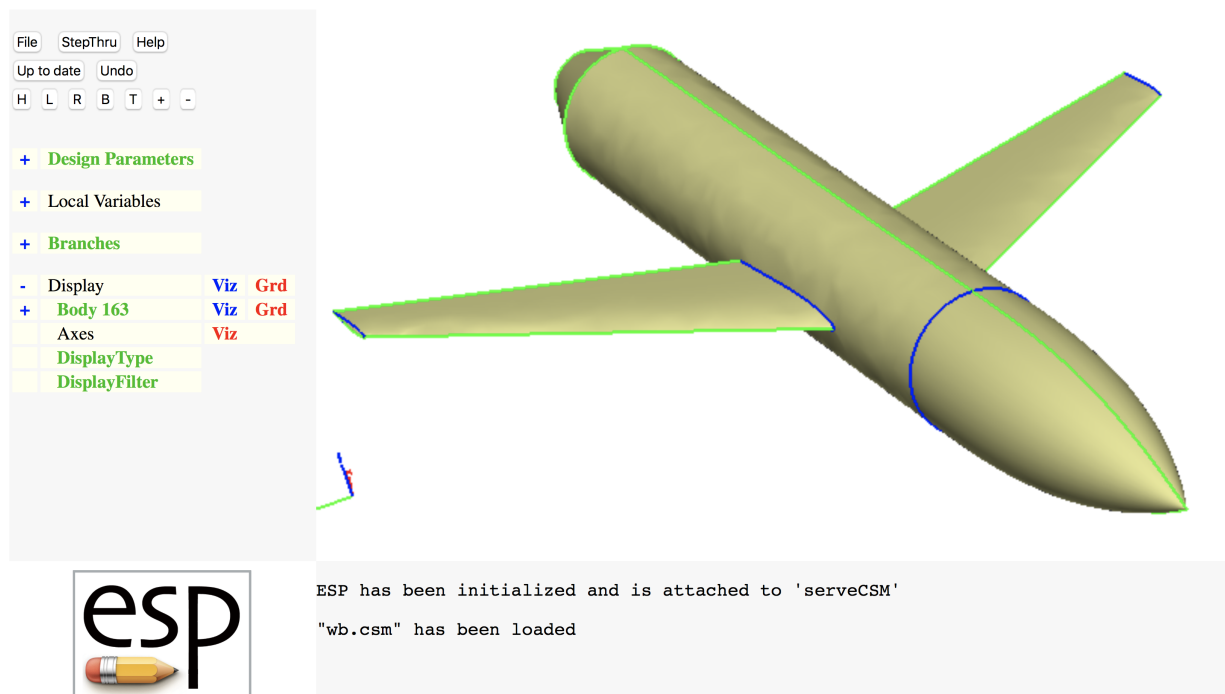
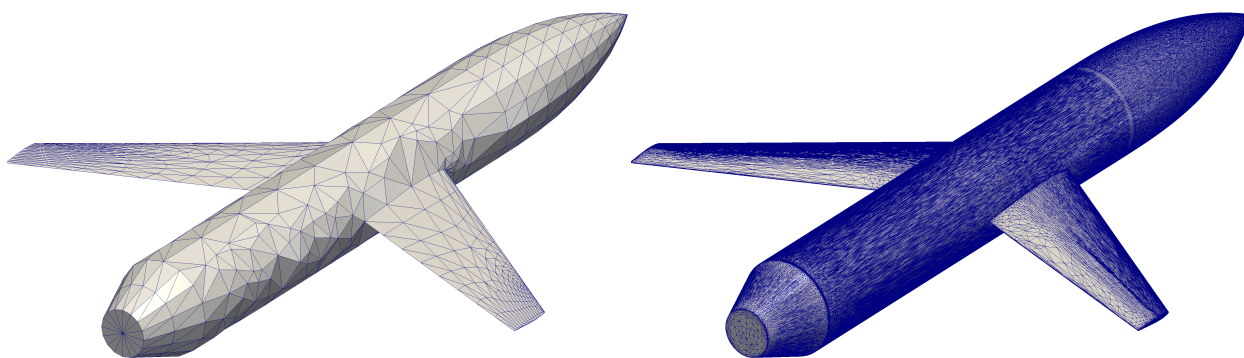


Fig. 8 AGARD wing body in ESP.



(a) Initial grid, 48K vertices.

(b) 20th grid, 1.8M vertices.

Fig. 9 AGARD wingbody surface grids.

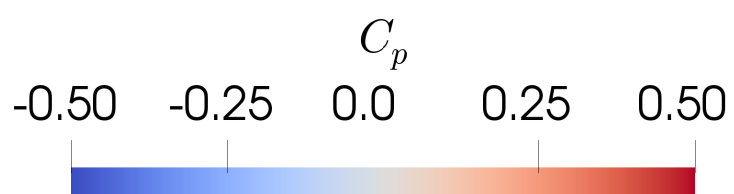
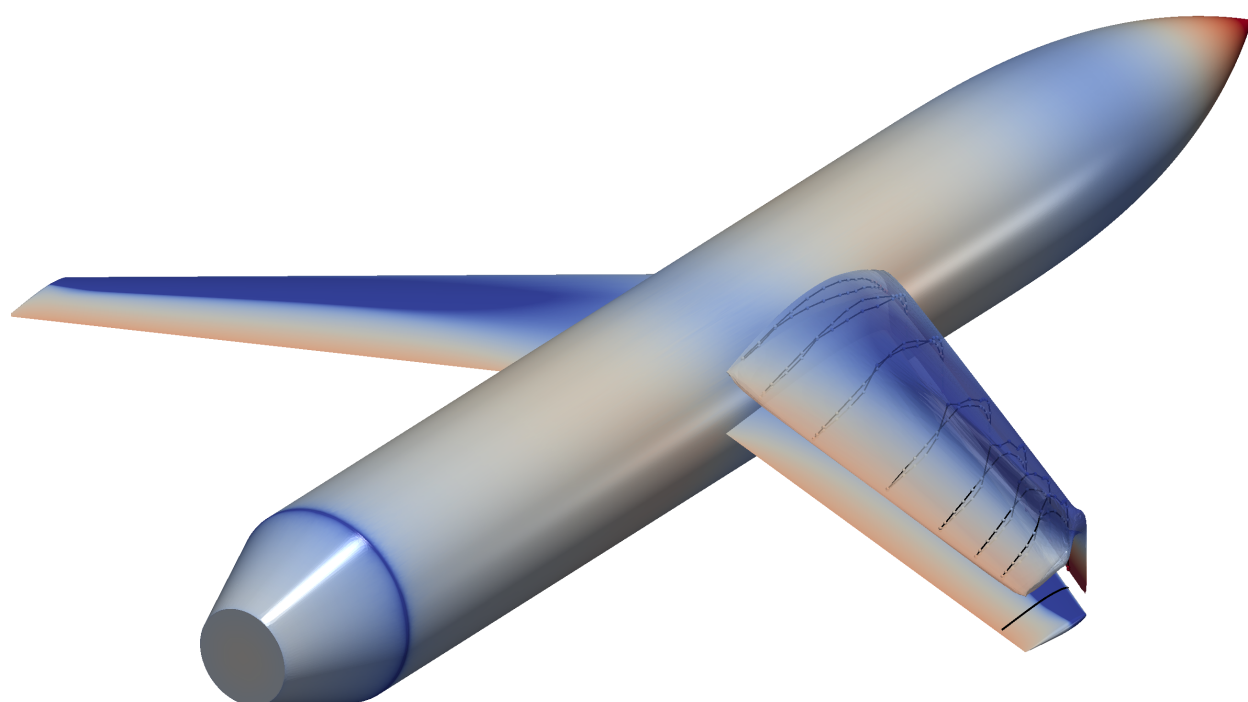


Fig. 10 AGARD wingbody surface pressures with FUN3D-FV+*refine*+EGADS, $M_\infty = 0.9$, $\alpha = 1^\circ$, $Re = 1 \times 10^6$.

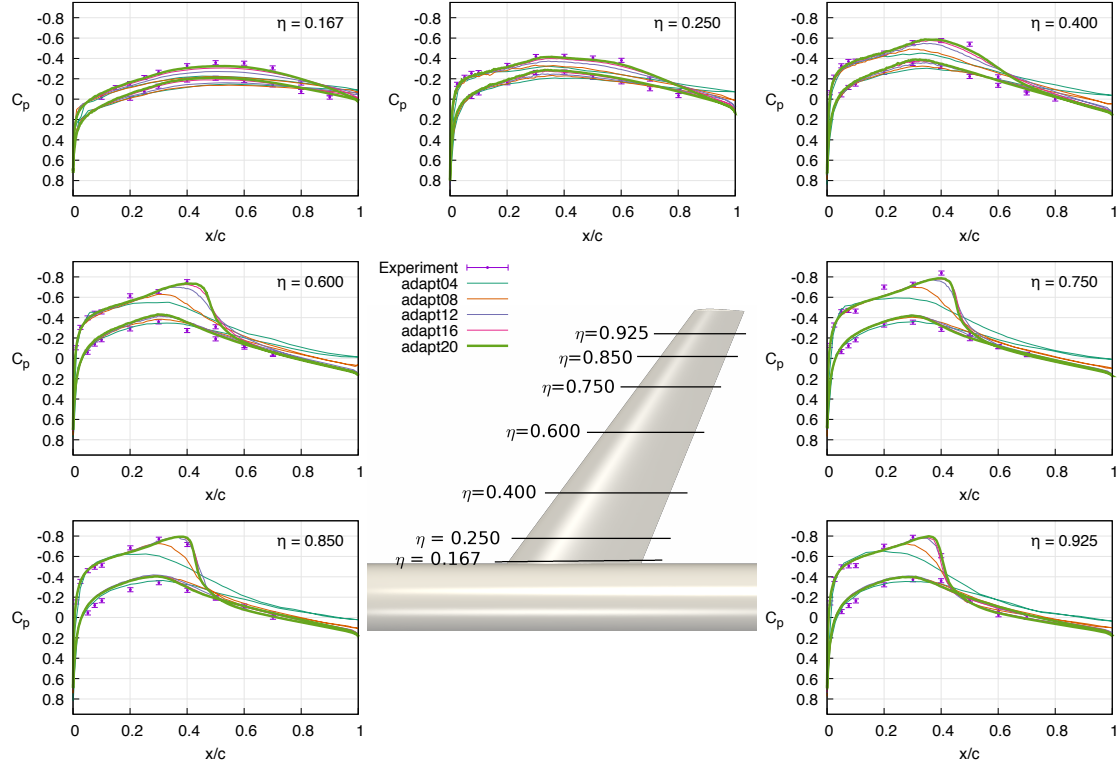


Fig. 11 AGARD wingbody wing pressure distributions with FUN3D-FV+refine+EGADS, $M_\infty = 0.9$, $\alpha = 1^\circ$, $Re = 1 \times 10^6$.

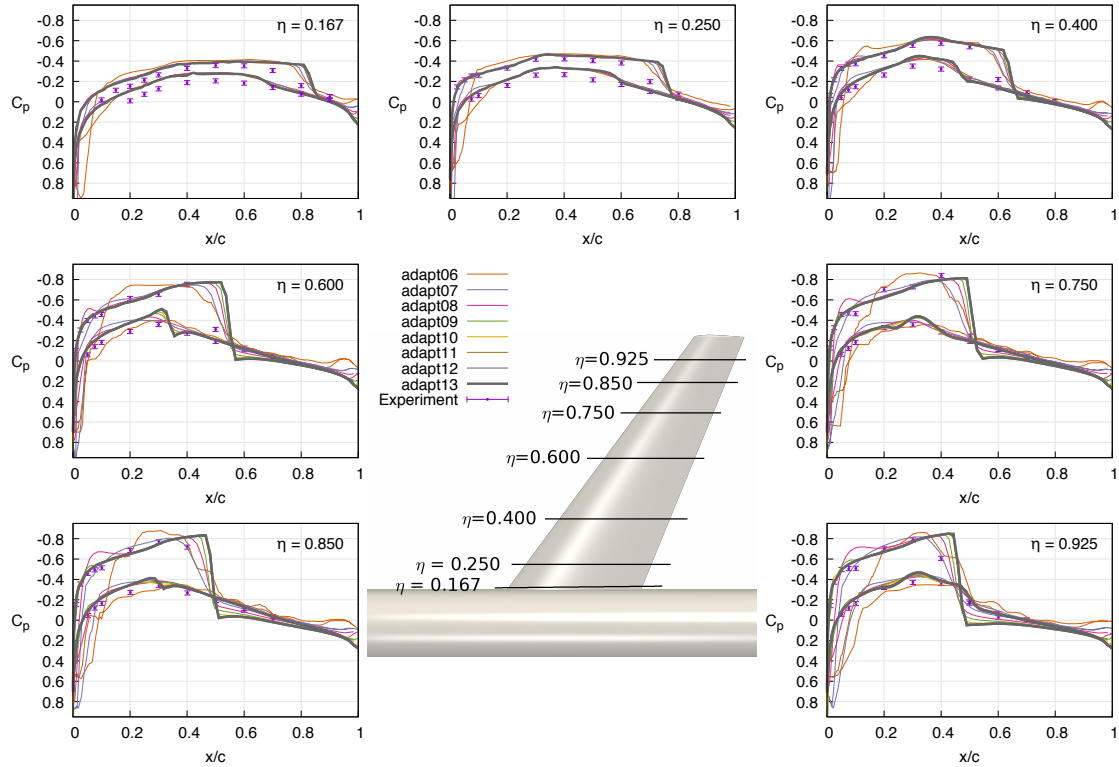


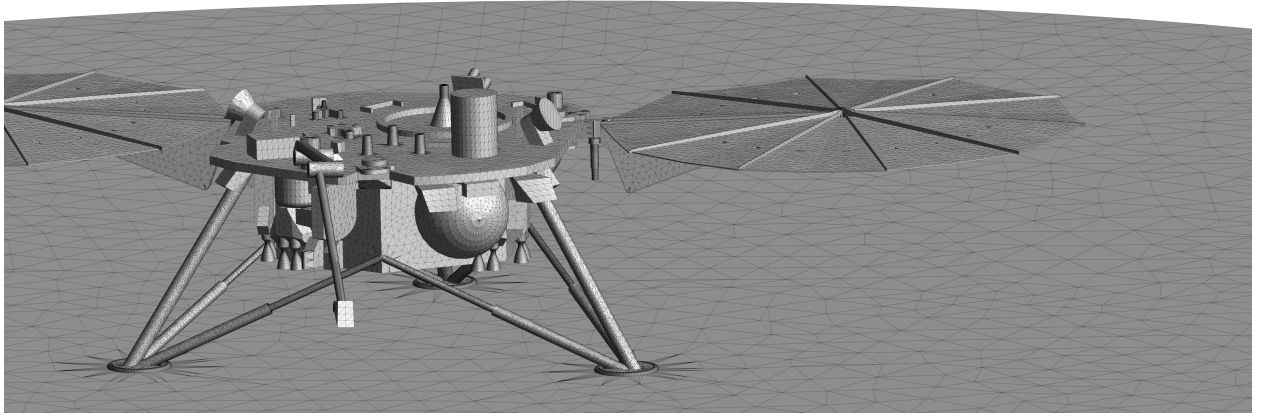
Fig. 12 AGARD wingbody wing pressure distributions with Cart3D adjoint, $M_\infty = 0.9$, $\alpha = 1^\circ$, $Re = 1 \times 10^6$.

D. Mars InSight Lander Ground Wind Loads

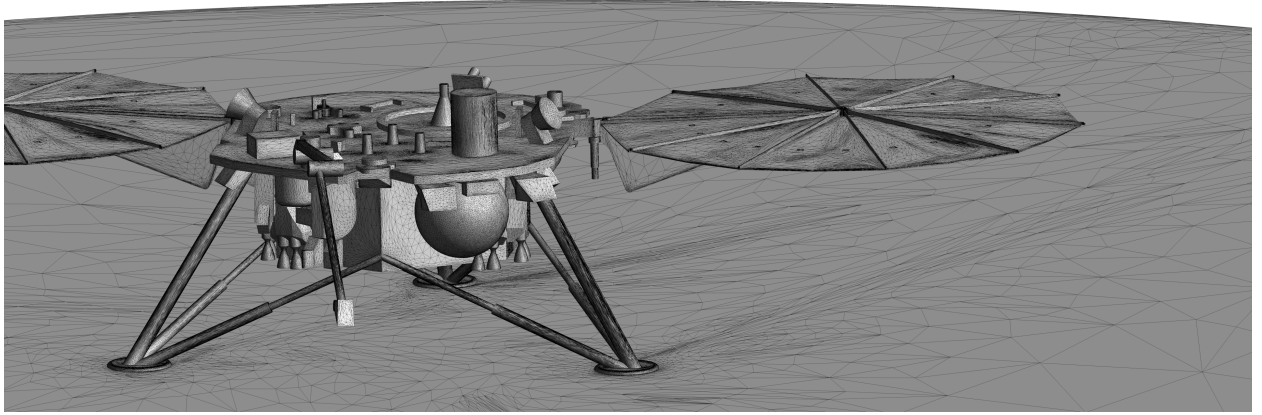
Seeking a more complex geometry (1800 faces) to test the workflow, Figure 13 depicts a ground-wind loads simulation for the Mars InSight Lander [65, 71, 72] where the flow direction is from left to right. Laminar flow at Mach 0.3, 30-degrees yaw, and a unit Reynolds number of 250K/m is simulated. The grid was adapted to control the L_2 interpolation error of the Mach field.

While the InSight lander geometry itself is in STEP format, the OpenCSM file that generated the computational domain is provided in Appendix C. EGADS provided the initial surface tessellation (Figure 13a) and TetGen provided the initial volume grid. The 29th surface grid in the automatic adaptive sequence is shown in Figure 13b, while Figure 14 shows a streamwise cut of Mach contours and the pressure coefficient on the lander and ground plane.

This complex geometry prompted two changes in the original sketch-to-solution workflow. First, robust treatment of periodic and degenerate geometry topologies; and second, the insertion of a curvature-based surface-grid adaptation of the raw tessellation received from EGADS to aid initial volume grid generation and subsequent grid adaptation. Both of these robustness enhancements are documented in Ref. [5].



(a) Initial grid; 679K vertices.



(b) 29th grid; 6M vertices.

Fig. 13 Mars InSight Lander ground wind loads simulation grids.

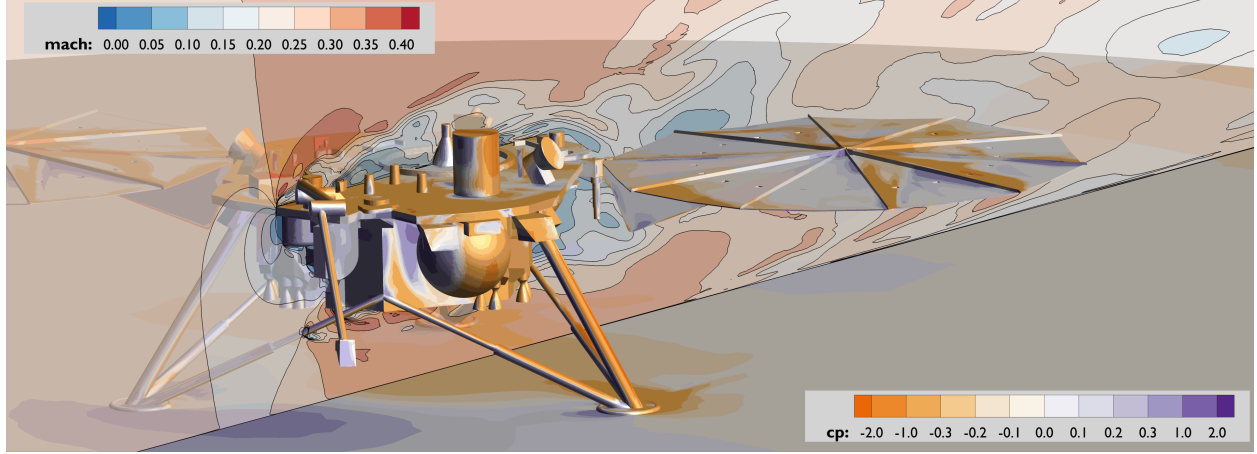


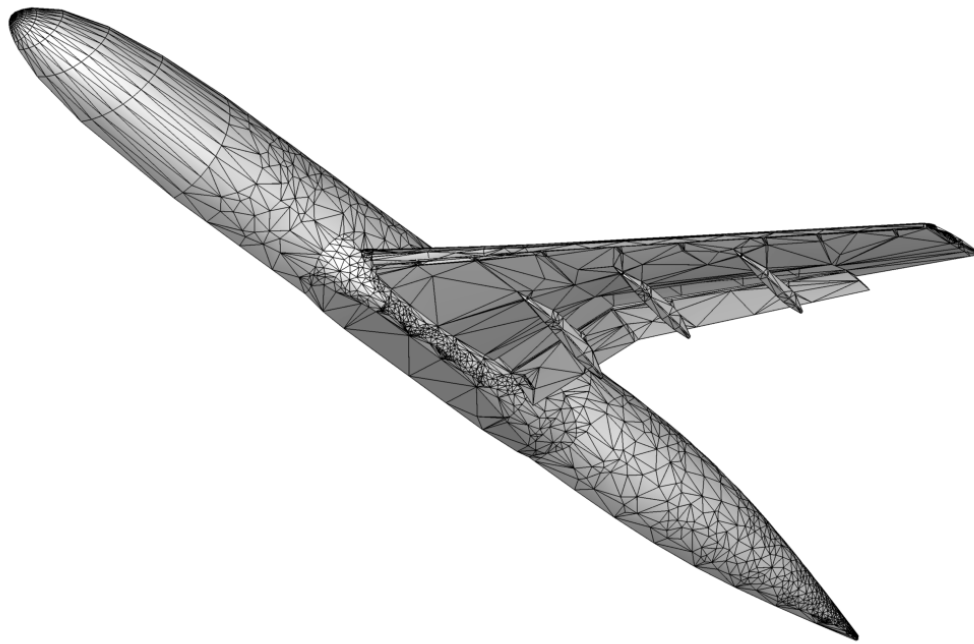
Fig. 14 Mach and pressure coefficient contours for Mars InSight Lander ground wind loads simulation: Mach 0.3, 30-degrees yaw, unit Reynolds number 250K/m.

E. Japan Aerospace Exploration Agency Standard Model

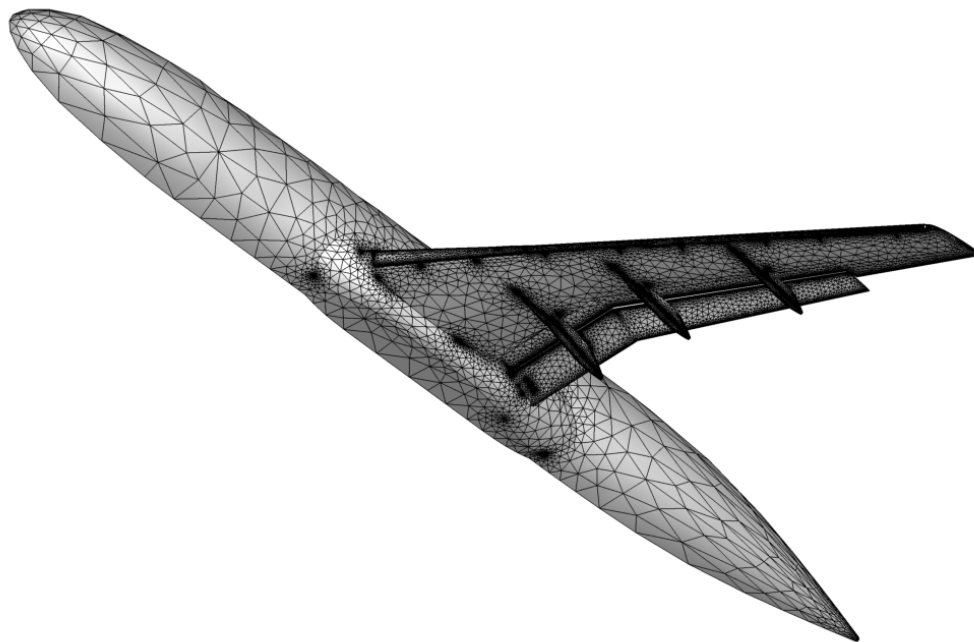
Japan Aerospace Exploration Agency Standard Model (JSM) with and without nacelle was part of the Third AIAA High Lift Prediction Workshop (HiLiftPW-3) [55]. Adaptive results for the nacelle-on and nacelle-off cases are shown in [73] and adaptive results for the nacelle-on case are shown in [5]. The Mach number is 0.172 for Case 2 of HiLiftPW-3. Reynolds Number based on mean aerodynamic chord is 1.93 million and the static temperature is 551.79 °R. The 10.47° angle of attack case of the drag polar is selected. The initial surface grid provided by EGADS and adapted to refine geometry constraints is shown in Figure 15. The high curvature and small features of the slat and flap are resolved in Figure 15b. The slat and flap attachment hardware are also resolved.

Force trajectories are shown in Figure 16. FUN3D-FV+*refine*+EGADS is shown with squares. These trajectories consist of a number of grid adaptations at fixed complexity (a small variation in h approximated by the number of vertices to the minus one-third power). The complexity is increased in a number of steps, which result in step decreases in h . The adapted grid forces are within the scatter of HiLiftPW-3 core participant submissions [55] shown as error bars. The conclusions of the HiLiftPW-3 summary and the range of values across the participant submissions indicate that the community has not obtained solutions and grids in the asymptotic region for this geometry and physical model. Accurate results for the JSM may require larger grids and/or output-based/goal-oriented metrics.

The JSM and symmetry plan colored with C_p contours are shown in Figure 17. The volume grid is sliced at $x = 3.5$ m constant fuselage station and the intersection of the tetrahedra and the plane is rendered. The thickness of boundary layer around the fuselage is resolved as well as the complex wake of the high lift devices and support brackets. The wing tip and flap end vortices are shown in the wing tip detail of Figure 17b. The flap support hardware creates an alternating set of vortices seen in the wake grid.

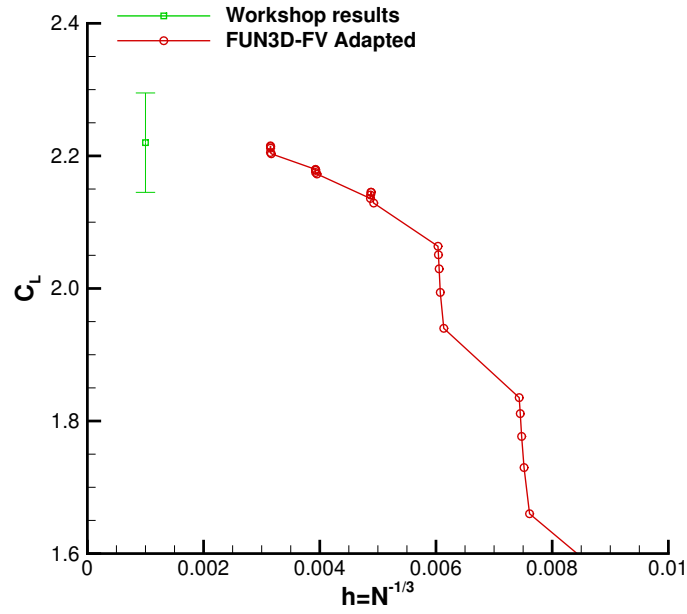


(a) Initial coarse EGADS tessellation.

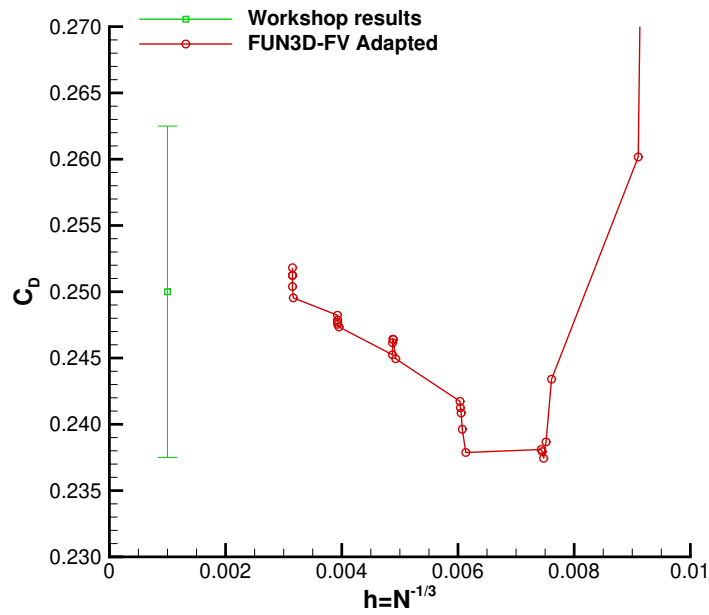


(b) Surface grid adapted to *refine* geometry constraints.

Fig. 15 Initial JSM surface grids.

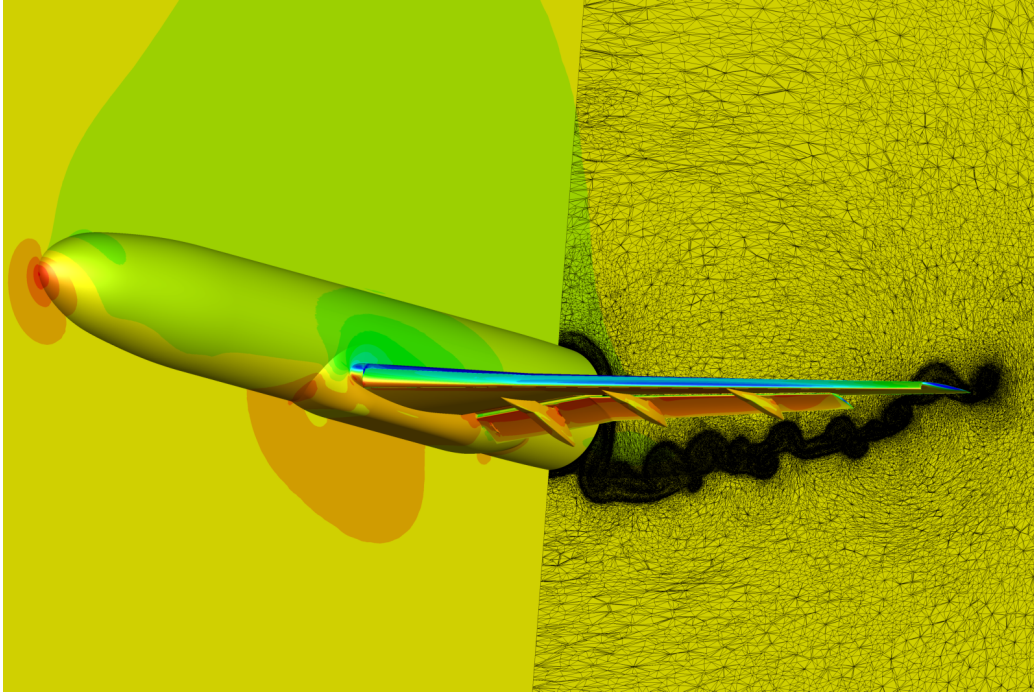


(a) Lift coefficient convergence.

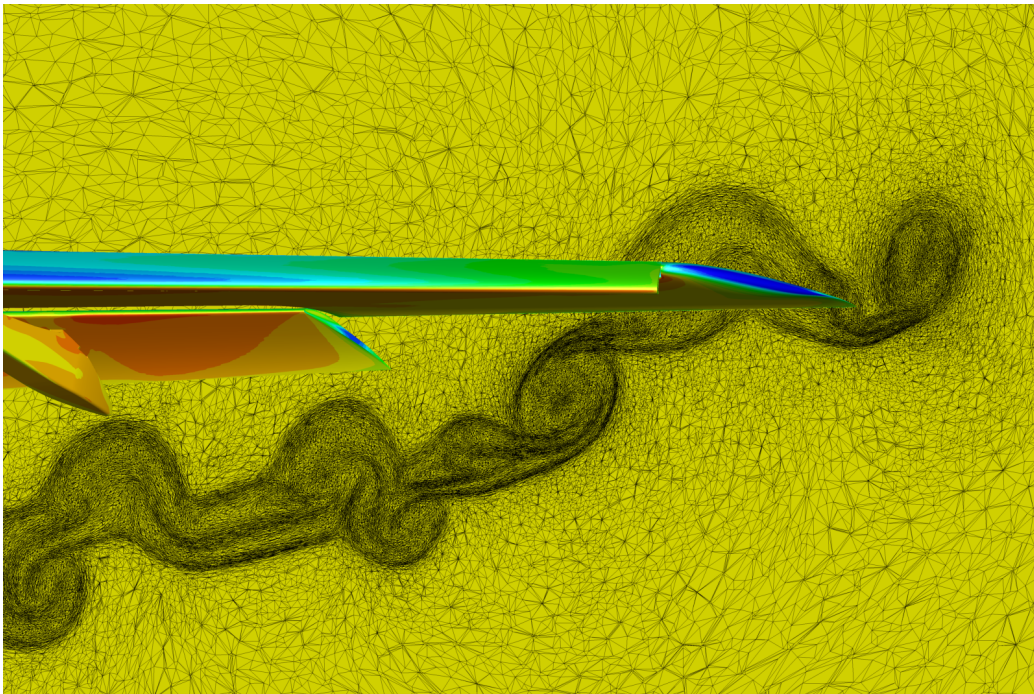


(b) Drag coefficient convergence.

Fig. 16 JSM force convergence trajectories.



(a) Semispan configuration and wake grid.



(b) Wing tip detail.

Fig. 17 JSM colored with C_p and $x = 3.5$ m slice of wing wake grid.

To study the ability to accommodate free shear layers, the Turbulence Modeling Resource (TMR) Axisymmetric Near-Sonic Jet (ANSJ) case[¶] was investigated. This TMR case models “Set Point 7” of an experimental campaign with the 1-inch radius Acoustic Research Nozzle 2 operated at a near-sonic jet exit Mach number (≈ 0.985) [53, 74]. The nozzle pressure and temperature ratios are 1.861 and 1.0, respectively. Figure 18 shows the domain and closeup of the nozzle. Unlike the TMR case, whose domain is a single-cell-wide wedge, this simulation domain consists of a one-quarter revolution bounded by y - and z -symmetry planes. The OpenCSM geometry model is provided in Appendix D.



(b) Detail view.

Figure 10 consists of two subplots. Subplot (a) shows a structured computational grid near a jet lip, with the x-axis ranging from -0.5 to 3.5 and the y-axis from 0 to 2. Subplot (b) displays Mach number contours for the same domain, with a color bar indicating Mach values from 0.1 to 1.1. The contours show a high-Mach region (red) near the jet lip and lower Mach regions (orange/yellow) further away.

For the second attempt, the Navier-Stokes equations were solved in a time-accurate manner. In this mode, the metric field used for grid adaptation is created from a series of snapshots taken over a period of time to capture the unsteadiness of the flow field [75, 76]. In this case, each grid was run for 2500 time steps (≈ 25 acoustic flow-through times) to flush any transients due to starting from the interpolated solution on the previous grid. Then the Mach field was saved every 10 time steps for 1000 time steps, and this was used to form the metric for the next grid adaptation. As shown in Figure 20, the grid follows the initial shear layer off the nozzle lip. While the jet shear layer spreading rate is better than the steady RANS-SA result, it is still too large as seen in Figure 21.

21

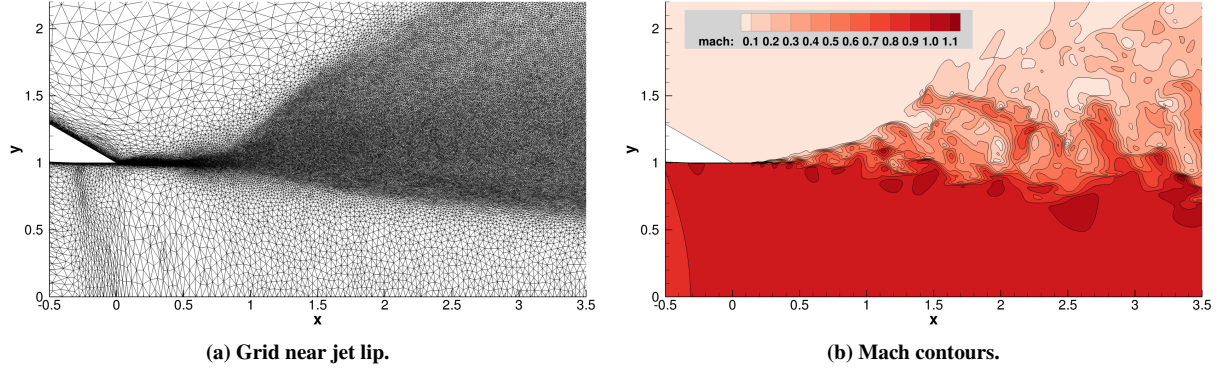


Fig. 20 Unsteady Navier-Stokes grid and Mach field after 42 grid adaptation cycles.

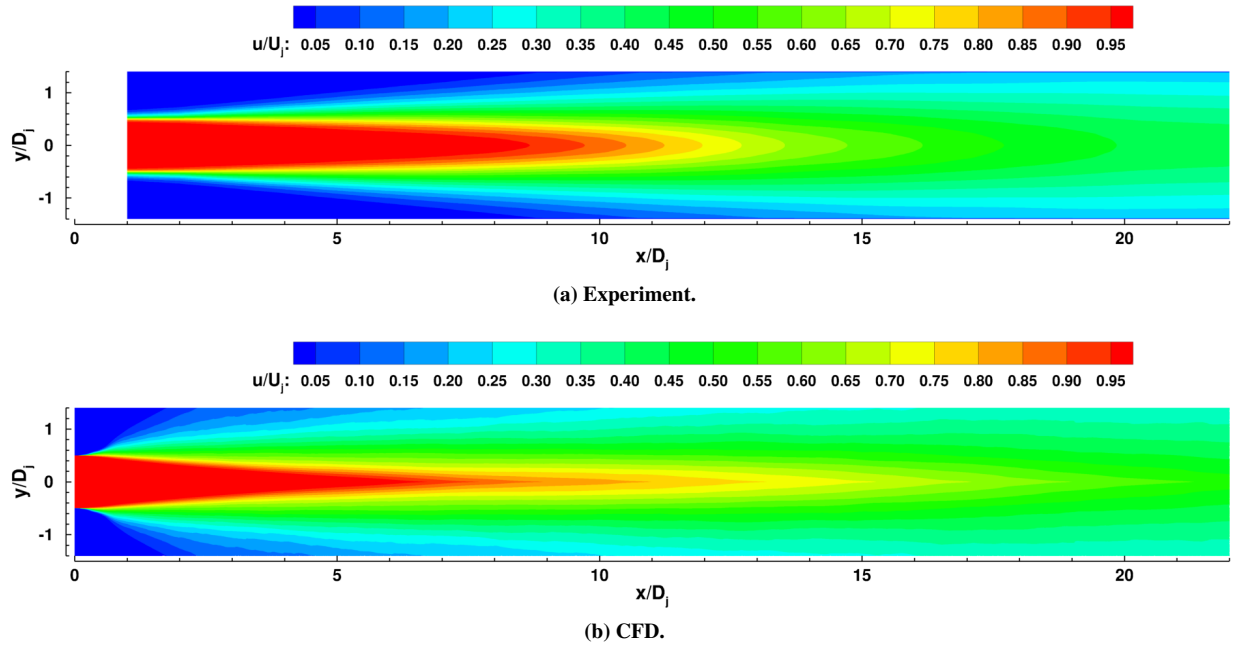


Fig. 21 Comparison of experiment [53] and CFD average u -velocity nondimensionalized by jet exit velocity.

IV. Concluding Remarks

This paper explored work flows to realize geometry-to-viscous CFD solution without an explicit human-in-the-loop grid generation step via grid adaptation driven by the L_2 norm of solution interpolation error. While we studied the application of this sketch-to-solution workflow for over 40 different cases, six were reported here to highlight issues uncovered along the way and to show the current state of the art. Some of the issues included initial grid generation, flow solver robustness, grid adaptation mechanics, geometry recovery, and error estimation. While the majority of the cases studied showed good agreement with experimental and workshop aerodynamic coefficients and pressure distributions, more detailed investigation is needed for second-order quantities like skin friction and heat rate.

Acknowledgments

The authors would like to acknowledge: John Dannenhoffer (Syracuse University) and Bob Haimes (MIT) for help with ESP, EGADS, and OpenCSM; Matt O’Connell (NASA Langley) for improving a load balance issue in Parfait’s distance calculator, reducing runtimes from hours to minutes; GeoLab’s Scott Brynildsen (Vigyan) and Norma Farr (Craig Technologies) for the Siemens NX[®] MCAD part of the Mars InSight Lander; and Jan-Reneé Carlson (NASA Langley) for help setting up the acoustic research nozzle case.

This work was supported by the Hypersonics Technology (HT) project of the Advanced Air Vehicles Program (AAVP) and the Transformational Tools and Technologies (TTT) project of the Transformative Aeronautics Concepts Program (TACP) under the NASA Aeronautics Research Mission Directorate (ARMD); the Space Launch System (SLS) and Orion Spacecraft programs under the NASA Human Exploration and Operations Mission Directorate (HEOMD); the Entry Systems Modeling project under the NASA Space Technology Mission Directorate (STMD); and the Research Directorate of NASA Langley Research Center.

References

- [1] Rogers, S. E., Dalle, D. J., and Chan, W. M., “CFD Simulations of the Space Launch System Ascent Aerodynamics and Booster Separation,” AIAA Paper 2015–778, 2015.
- [2] Alauzet, F., and Loseille, A., “A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics,” *Computer-Aided Design*, Vol. 72, 2016, pp. 13–39. doi:10.1016/j.cad.2015.09.005, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.
- [3] Ibanez, D., Barral, N., Krakos, J., Loseille, A., Michal, T., and Park, M., “First Benchmark of the Unstructured Grid Adaptation Working Group,” *Procedia Engineering*, Vol. 203, 2017, pp. 154–166. doi:10.1016/j.proeng.2017.09.800, 26th International Meshing Roundtable, IMR26, 18–21 Sept. 2017, Barcelona, Spain.
- [4] Park, M. A., Balan, A., Anderson, W. K., Galbraith, M. C., Caplan, P. C., Carson, H. A., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., “Verification of Unstructured Grid Adaptation Components,” AIAA Paper 2019–1723, 2019.
- [5] Park, M. A., Kleb, B., Jones, W. T., Krakos, J. A., Michal, T., Loseille, A., Haimes, R., and Dannenhoffer, J. F., III, “Geometry Modeling for Unstructured Mesh Adaptation,” AVIATION Forum 2019, American Institute of Aeronautics and Astronautics, Reston, VA (submitted for publication).
- [6] Taylor, N. J., and Haimes, R., “Geometry Modelling: Underlying Concepts and Requirements for Computational Simulation,” AIAA Paper 2018–3402, 2018.
- [7] Chawner, J. R., Michal, T., Slotnick, J. P., and Rumsey, C. L., “Summary of the 1st AIAA Geometry and Mesh Generation Workshop (GMGW-1) and Future Plans,” AIAA Paper 2018–128, 2018.
- [8] Park, M. A., Krakos, J. A., Michal, T., Loseille, A., and Alonso, J. J., “Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Toward CFD Vision 2030,” AIAA Paper 2016–3323, 2016.
- [9] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences,” NASA CR-2014-218178, Langley Research Center, Mar. 2014. doi:2060/20140003093.
- [10] Gibson, W., NPR Fresh Air interview, Aug. 1993.

- [11] Nemec, M., and Aftosmis, M. J., “Toward Automatic Verification of Goal-Oriented Flow Simulations,” NASA TM-2014-218386, Ames Research Center, Aug. 2014. doi:2060/20150000864.
- [12] K  nig, B., and Fares, E., “Validation of a Transonic Lattice-Boltzmann Method on the NASA Common Research Model,” AIAA Paper 2016–2023, 2016.
- [13] Holman, D. M., Brionnaud, R. M., Modena, M. C., and S  nchez, E. V., “Lattice Boltzmann Method Contribution to the Second High-Lift Prediction Workshop,” *AIAA Journal of Aircraft*, Vol. 52, No. 4, 2015, pp. 1122–1135.
- [14] Barad, M. F., Kocheemoolayil, J. G., and Kiris, C. C., “Lattice Boltzmann and Navier-Stokes Cartesian CFD Approaches for Airframe Noise Predictions,” AIAA Paper 2017–4404, 2017.
- [15] Demargne, A. A. J., Evans, R. O., Tiller, P. J., and Dawes, W. N., “Practical and Reliable Mesh Generation for Complex, Real-world Geometries,” AIAA Paper 2014–119, 2014.
- [16] Dawes, B., Kellar, W., Verdicchio, J., Racz, T., Meah, N., Kudryavtsev, A., Evans, R., Hunt, M., Tiller, P., Vasco, C., Yi, L., and Liu, K., “Contributions Towards the NASA 2030 Vision for Simulation,” AIAA Paper 2018–1499, 2018.
- [17] *Oxford English Dictionary*, Oxford University Press, 2018.
- [18] Haimes, R., and Dannenhoffer, J. F., III, “The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry,” AIAA Paper 2013–3073, 2013.
- [19] Dannenhoffer, J. F., III, “OpenCSM: An Open-Source Constructive Solid Modeler for MDAO,” AIAA Paper 2013–701, 2013.
- [20] Haimes, R., and Drela, M., “On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design,” AIAA Paper 2012–683, 2012.
- [21] Marcum, D. L., and Weatherill, N. P., “Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection,” *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625. doi:10.2514/3.12701.
- [22] Si, H., “TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator,” *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 2, 2015, pp. 11:1–11:36. doi:10.1145/2629697.
- [23] Anderson, W. K., and Bonhaus, D. L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X.
- [24] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., “FUN3D Manual: 13.4,” NASA TM-2018-220096, Langley Research Center, Oct. 2018. doi:2060/20180007519.
- [25] Nielsen, E. J., Lu, J. C.-C., Park, M. A., and Darmofal, D. L., “An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids,” *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155.
- [26] Spalart, P. R., and Allmaras, S. R., “A One-Equation Turbulence Model for Aerodynamic Flows,” *La Recherche A  rospatiale*, Vol. 1, 1994, pp. 5–21.
- [27] “Parfait Toolkit,” <https://github.com/T-infinity/parfait>, Last accessed April, 2019.
- [28] Bonet, J., and Peraire, J., “An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems,” *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17.
- [29] Barth, T. J., “A 3-D Upwind Euler Solver for Unstructured Meshes,” AIAA Paper 91–1548, 1991.
- [30] Nishikawa, H., “Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes,” AIAA Paper 2010–5093, 2010.
- [31] O’Connell, M. D., Druyor, C. T., Thompson, K. B., Jacobson, K. E., Anderson, W. K., Nielsen, E. J., Carlson, J.-R., Park, M. A., Jones, W. T., Biedron, R. T., Kleb, B., and Zhang, X., “T-infinity: The Dependency Inversion Principle for Rapid and Sustainable Multidisciplinary Software Development,” AIAA Paper 2018–3856, 2018.
- [32] Abhyankar, S., Brown, J., Constantinescu, E. M., Ghosh, D., Smith, B. F., and Zhang, H., “PETSc/TS: A Modern Scalable ODE/DAE Solver Library,” *Computing Research Repository (CoRR)*, Vol. Numerical Analysis (math.NA), No. arXiv:1806.01437, 2018. URL <http://arxiv.org/abs/1302.6066>.

- [33] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., “PETSc Users Manual,” Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.
- [34] Thompson, K. B., and O’Connell, M. D., “Streamlined Convergence Acceleration for CFD Codes,” AVIATION Forum 2019, American Institute of Aeronautics and Astronautics, Reston, VA (submitted for publication).
- [35] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., “Adaptive Remeshing for Compressible Flow Computations,” *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466. doi:10.1016/0021-9991(87)90093-3.
- [36] Peraire, J., Peirò, J., and Morgan, K., “Adaptive Remeshing for Three-Dimensional Compressible Flow Computations,” *Journal of Computational Physics*, Vol. 103, No. 2, 1992, pp. 269–285. doi:10.1016/0021-9991(92)90401-J.
- [37] Alauzet, F., and Loseille, A., “High-Order Sonic Boom Modeling Based on Adaptive Methods,” *Journal of Computational Physics*, Vol. 229, No. 3, 2010, pp. 561–593. doi:10.1016/j.jcp.2009.09.020.
- [38] Alauzet, F., “Size Gradation Control of Anisotropic Meshes,” *Finite Elements in Analysis and Design*, Vol. 46, No. 1–2, 2010, pp. 181–202. doi:10.1016/j.finel.2009.06.028.
- [39] Loseille, A., and Alauzet, F., “Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error,” *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60. doi:10.1137/090754078.
- [40] Haimes, R., and Dannenhoffer, J. F., III, “EGADSLite: A Lightweight Geometry Kernel for HPC,” AIAA Paper 2018–1401, 2018.
- [41] Park, M. A., “Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows,” Ph.D. thesis, Massachusetts Institute of Technology, Sep. 2008. doi:1721.1/46363.
- [42] Loseille, A., and Löhner, R., “Cavity-Based Operators for Mesh Adaptation,” AIAA Paper 2013–152, 2013.
- [43] Alauzet, F., “A Changing-Topology Moving Mesh Technique for Large Displacements,” *Engineering with Computers*, Vol. 30, No. 2, 2014, pp. 175–200. doi:10.1007/s00366-013-0340-z.
- [44] Freitag, L. A., and Ollivier-Gooch, C., “Tetrahedral Mesh Improvement Using Swapping and Smoothing,” *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 3979–4002. doi:10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.
- [45] Gnemmi, P., Adeli, R., and Longo, J., “Computational Comparisons of the Interaction of a Lateral Jet on a Supersonic Generic Missile,” AIAA Paper 2008–6883, 2008.
- [46] Chu, J., and Luckring, J. M., “Experimental Surface Pressure Data Obtained on 65 deg Delta Wing Across Reynolds Number and Mach Number Ranges. Volume 1–Sharp Leading Edge,” NASA TM-4645-Vol-1, Langley Research Center, Feb. 1996. doi:2060/19960025648.
- [47] Dansberry, B. E., Durham, M. H., Bennett, R. M., Turnock, D. L., Silva, W. A., and Rivera Jr., J. A., “Physical Properties of the Benchmark Models Program Supercritical Wing,” NASA TM-4457, Langley Research Center, Sep. 1993. doi:2060/19940011249.
- [48] Chwalowski, P., Quon, E., and Brynildsen, S. E., “Computational Analysis of the Transonic Dynamics Tunnel Using FUN3D,” AIAA Paper 2016–1775, 2016.
- [49] Hofferth, J. W., and Ogg, D. R., “Reactivation of VKF Wind Tunnel D by AFRL at AEDC: Overview and Subsystem Checkout Results,” AIAA Paper 2018-5263, 2018. doi:10.2514/6.2018-5263.
- [50] Schmitt, V., and Charpin, F., “Pressure Distributions on the ONEAR-M6-Wing at Transonic Mach Numbers,” *Experimental Data Base for Computer Program Assessment: Report of the Fluid Dynamics Panel Working Group 04*, AR-138, NATO Research and Technology Organisation AGARD, 1979, pp. B1:1–B1:44.
- [51] Park, M. A., and Nemec, M., “Near Field Summary and Statistical Analysis of the Second AIAA Sonic Boom Prediction Workshop,” *AIAA Journal of Aircraft*, 2018. doi:10.2514/1.C034866.
- [52] Rumsey, C. L., “The NASA Juncture Flow Test as a Model for Effective CFD/Experimental Collaboration,” AIAA Paper 2018–3319, 2018.

- [53] Bridges, J., and Wernet, M. P., "Establishing Consensus Turbulence Statistics for Hot Subsonic Jets," AIAA Paper 2010-3751, June 2010.
- [54] Rivers, M. B., "NASA Common Research Model: A History and Future Plans," AIAA Paper 2019-2188, 2019.
- [55] Rumsey, C. L., Slotnick, J. P., and Sclafani, A. J., "Overview and Summary of the Third AIAA High Lift Prediction Workshop," *AIAA Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 621–644. doi:10.2514/1.C034940.
- [56] Durston, D. A., Cliff, S. E., Denison, M. F., Jensen, J. C., Moran, P. J., Smith, N. T., Heineck, J. T., Schairer, E. T., Kushner, L. K., Castner, R. S., Elmilgui, A. A., Carter, M. B., Winski, C. S., and Shea, P. R., "Nozzle Plume/Shock Interaction Experimental and Computational Sonic Boom Analyses from the NASA Ames 9- by 7-Foot Supersonic Wind Tunnel," NASA TP-2018-219879, Dec. 2018. doi:2060/20190001341.
- [57] Treadgold, D. A., Jones, A. F., and Wilson, K. H., "Pressure Distribution Measured in the RAE 8 ft x 6 ft Transonic Wind Tunnel on RAE Wing 'A' in Combination with an Axi-Symmetric Body at Mach numbers of 0.4, 0.8, and 0.9," *Experimental Data Base for Computer Program Assessment: Report of the Fluid Dynamics Panel Working Group 04*, AR-138, NATO Research and Technology Organisation AGARD, 1979, pp. B4:1–B4:44.
- [58] Horvath, T. J., Berry, S. A., Merski, N. R., and Fitzgerald, S. M., "X-38 Experimental Aerothermodynamics," *Journal of Spacecraft and Rockets*, Vol. 41, No. 2, 2004, pp. 272–292. doi:10.2514/1.9198.
- [59] Bobbitt, P. J., Maglieri, D. J., Banks, D. W., Frederick, M. A., and Fuchs, A. W., "Wind-Tunnel and Flight Test Results for the Measurements of Flow Variables at Supersonic Speeds Using Improved Wedge and Conical Probes," NASA TM-2012-216004, 2012. doi:2060/20130001760.
- [60] Dillman, R. A., DiNonno, J. M., Bodkin, R. J., Hughes, S. J., Cheatwood, F. M., Blakeley, H., Akamine, R. L., and Bowes, A., "Planned Orbital Flight Test of a 6 m HIAD," Presentation, International Planetary Probe Workshop, Jun. 2018.
- [61] Jackson, C. M., Jr., Corlett, W. A., and Monta, W. J., "Description and Calibration of the Langley Unitary Plan Wind Tunnel," NASA TP-1905, Langley Research Center, Nov. 1981. doi:2060/19820004162.
- [62] Riebe, G. D., and Fox, C. H., Jr., "Subsonic Maneuver Capability of a Supersonic Cruise Fighter Wing Concept," NASA TP-2642, Langley Research Center, Jan. 1987. doi:2060/19870005751.
- [63] Ware, G. M., and Cruz, C. I., "Aerodynamic Characteristics of the HL-20," *Journal of Spacecraft and Rockets*, Vol. 30, No. 5, 1993, pp. 529–536. doi:10.2514/3.25562.
- [64] Oberkampf, W. L., and Aeschliman, D. P., "Joint Computational/Experimental Aerodynamic Research on a Hypersonic Vehicle, Part 1: Experimental Results," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2000–2009. doi:10.2514/3.11172.
- [65] Abilleira, F., Halsell, A., Chung, M.-K., Fujii, K., Gustafson, E., Hahn, Y., Lee, J., McPheeters, S.-E., Mottinger, N., Seubert, J., Sklyanskiy, E., and Wallace, M., "2018 Mars Insight Mission Design and Navigation Overview," AAS 18-251, AAS/AIAA Astrodynamics Specialist Conference, 2018. doi:2014/45975.
- [66] McWherter Walker, M., and Oberkampf, W. L., "Joint Computational/Experimental Aerodynamic Research on a Hypersonic Vehicle, Part 2: Computational Results," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2010–2016. doi:10.2514/3.11173.
- [67] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-Order CFD Methods: Current Status and Perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845. doi:10.1002/fld.3767.
- [68] Klaij, C. M., van der Vegt, J. J. W., and van der Ven, H., "Space-time Discontinuous Galerkin Method for the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 217, No. 2, 2006, pp. 589–611. doi:10.1016/j.jcp.2006.01.018.
- [69] Riley, A. J., and Lowson, M. V., "Development of a Three-Dimensional Free Shear Layer," *Journal of Fluid Mechanics*, Vol. 369, 1998, pp. 49–89.
- [70] Nemec, M., and Aftosmis, M. J., "Adjoint Sensitivity Computations for an Embedded-Boundary Cartesian Mesh Method," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2724–2742. doi:10.1016/j.jcp.2007.11.018.
- [71] Abilleira, F., Frauenholz, R., Fujii, K., Wallace, M., and You, T.-H., "Mars InSight Mission Design and Navigation," AAS 14-363, 4th AAS/AIAA Space Flight Mechanics Meeting, 2014. doi:2014/45459.
- [72] Lam, G. Q., Billets, S. A., Norick, T. A., and Warwick, R. W., "Solar Array Design for the Mars InSight Lander Mission," AIAA Paper 2016-4520, 2016.

- [73] Michal, T., Kamenetskiy, D., and Krakos, J., “Anisotropic Adaptive Mesh Results for the Third High Lift Prediction Workshop (HiLiftPW-3),” AIAA Paper 2018–1257, 2018.
- [74] Bridges, J., and Wernet, M. P., “The NASA Subsonic Jet Particle Image Velocimetry (PIV) Dataset,” NASA/TM 2011-216807, November 2011.
- [75] Barral, N., “Time-accurate anisotropic mesh adaptation for three-dimensional moving mesh problems,” Ph.D. thesis, Université Pierre et Marie Curie - Paris VI, Nov. 2015.
- [76] Alauzet, F., Frey, P. J., George, P.-L., and Mohammadi, B., “3D Transient Fixed Point Mesh Adaptation for Time-Dependent Problems: Application to CFD Simulations,” *Journal of Computational Physics*, Vol. 222, No. 2, 2007, pp. 592–623. doi:10.1016/j.jcp.2006.08.012.

Appendices

OpenCSM geometry files used by some of the cases explored above.

A. OpenCSM Geometry Model for JCEAP Sliced Sphere Cone with Flap

```
# Sandia sphere-cone with slice tested by Oberkampf

# Constant and Design Parameters:

# Sharp-cone length in feet
despmtr  virtualLength  0.94525

# Cone half-angle
despmtr  halfAngle      10.00000

# Nose-sphere radius to cone-base radius ratio
despmtr  nosetobase     0.1000

# Fraction of base radius (or cone length) to slice the flat
despmtr  sliceFraction  0.72520

# Width of body flap as ratio to cone base radius
despmtr  flapWidthFraction  1.09400

# Body flap deflection angle, degrees
despmtr  flapAngle      20.00000

set      baseRadius virtualLength*tand(halfAngle)
set      noseRadius  nosetobase*baseRadius
set      halfFlap    flapWidthFraction/2
set      hinge        virtualLength*sqrt(sliceFraction*sliceFraction+halfFlap*halfFlap)

# Farfield
sphere   0.75*virtualLength  0  0  virtualLength

# Begin with sharp cone
cone     0  0  0  virtualLength  0  0  baseRadius

# Round off the spherical nose cap
cylinder 0  0  0  -virtualLength  0  0  noseRadius*cosd(halfAngle)
sphere   0  0  0  noseRadius
subtract none 1  0
translate noseRadius/sind(halfAngle)  0  0
subtract none 1  0

# Slice off the longitudinal flat
box      0  sliceFraction*baseRadius  -baseRadius  1.1*virtualLength  baseRadius  2*baseRadius
subtract none 1  0

# Create body flap
box      hinge 0  -halfFlap*baseRadius  3*(virtualLength-hinge)  sliceFraction*baseRadius  flapWidthFraction*baseRadius
rotatez  flapAngle  hinge  sliceFraction*baseRadius
union    0  0  0

# Chop off debris out the aft end of the model
cylinder virtualLength  0  0  1.5*virtualLength  0  0  2*baseRadius
subtract none 1  0

# Rotate into FUN3D coordinates, +z is up
rotatex  -90  0  0
```

```
# Remove the model from the flow domain
subtract none 1 0

# Emit an EGADS file that refine can use
dump sandiacone.egads

end
```

B. OpenCSM Geometry Model for AGARD Wingbody Case

```
# AGARD AR-138 Wing A Body B2
# 'Experimental Data Base for Computer Program Assessment',
# AGARD, 1979
#
# Geometry and experimental data:
# D.A. Treadgold, A.F. Jones, and K.H. Wilson, "Pressure
# Distribution Measured in the RA 8ft x 6ft Transonic Wind
# Tunnel on RAE Wing 'A' in Combination with an Axi-
# Symmetric Body at Mach Numbers of 0.4, 0.8 and 0.9,"
# Appendix B4 of the AGARD-AR-138, EXPERIMENTAL DATA BASE
# FOR COMPUTER PROGRAM ASSESSMENT, Report of the Fluid
# Dynamics Panel Working Group 04 (Available from National
# Technical Information Service (NTIS), 5285 Port Royal Road,
# Springfield, VA 22161).

despmtr diameter 6.0
despmtr fuselage_start 16.0
despmtr length 45.95
despmtr boattail_len 4.00
despmtr boattail_base 3.00
despmtr root_chord 9.00
despmtr root_xloc 20.00
despmtr tip_chord 3.00
despmtr wing_span 18.00
despmtr wing_le_sweep 36.65
despmtr nose_length 16.00
despmtr nose_spline_pts 20
despmtr generate_farfield 1

# make nose

skbeg nose_length 0. -diameter/2.
linseg nose_length 0. 0.
linseg 0 0. 0.
set divs nose_spline_pts
patbeg s divs
set x s/divs
spline x*nose_length \
0 \
-(16/3)*(0.5*x-0.375*x^2+(1/16)*x^4)*diameter/2
patend
skend
revolve 0 0 0 1 0 0 180
store nose
restore nose
mirror 0 1 0
restore nose
union
box 0.17936 0.5 0.5 -1 -1 -1
subtract
sphere 0.2251 0. 0. 0.1
union

skbeg fuselage_start 0 0
linseg length 0 0
linseg length+boattail_len 0 0
linseg length+boattail_len 0 boattail_base/2
linseg length 0 diameter/2
linseg fuselage_start 0 diameter/2
linseg fuselage_start 0 0
skend

# create fuselage and boat tail

revolve fuselage_start 0 0 1 0 0 180
store fuselage
restore fuselage
mirror 0 1 0

restore fuselage
join

union

# create wing from RAE 'A' airfoil

skbeg 1.0000 0.000000 0
spline 1.0000 0.000000 0
spline 0.9800 0.001610 0
spline 0.9600 0.003219 0
spline 0.9400 0.004829 0
spline 0.9200 0.006439 0
spline 0.8800 0.009658 0
spline 0.8500 0.012073 0
spline 0.8200 0.014487 0
spline 0.7800 0.017707 0
spline 0.7500 0.020121 0
spline 0.7200 0.022531 0
spline 0.6800 0.025707 0
spline 0.6500 0.028039 0
spline 0.6200 0.030308 0
spline 0.5800 0.033209 0
spline 0.5500 0.035265 0
spline 0.5200 0.037196 0
spline 0.4800 0.039539 0
spline 0.4500 0.041091 0
spline 0.4200 0.042438 0
spline 0.3800 0.043855 0
spline 0.3500 0.044582 0
spline 0.3200 0.044960 0
spline 0.2800 0.044730 0
spline 0.2500 0.043966 0
spline 0.2200 0.042746 0
spline 0.1800 0.040380 0
spline 0.1500 0.037982 0
spline 0.1200 0.034938 0
spline 0.0900 0.031067 0
spline 0.0750 0.028722 0
spline 0.0600 0.026008 0
spline 0.0400 0.021577 0
spline 0.0300 0.018832 0
spline 0.0200 0.015494 0
spline 0.0160 0.013901 0
spline 0.0125 0.012318 0
spline 0.0100 0.011039 0
spline 0.0080 0.009888 0
spline 0.0070 0.009256 0
spline 0.0050 0.007835 0
spline 0.0030 0.006078 0
spline 0.0010 0.003515 0
spline 0.0000 0.000000 0
linseg 0.0000 0.000000 0
spline 0.0010 -0.003515 0
spline 0.0030 -0.006078 0
spline 0.0050 -0.007835 0
spline 0.0070 -0.009256 0
spline 0.0080 -0.009888 0
spline 0.0100 -0.011039 0
spline 0.0125 -0.012318 0
spline 0.0160 -0.013901 0
spline 0.0200 -0.015494 0
spline 0.0300 -0.018832 0
spline 0.0400 -0.021577 0
spline 0.0600 -0.026008 0
spline 0.0750 -0.028722 0
spline 0.0900 -0.031067 0
spline 0.1200 -0.034938 0
```



```

spline 0.1500 -0.037982 0
spline 0.1800 -0.040380 0
spline 0.2200 -0.042746 0
spline 0.2500 -0.043966 0
spline 0.2800 -0.044730 0
spline 0.3200 -0.044960 0
spline 0.3500 -0.044582 0
spline 0.3800 -0.043855 0
spline 0.4200 -0.042438 0
spline 0.4500 -0.041091 0
spline 0.4800 -0.039539 0
spline 0.5200 -0.037196 0
spline 0.5500 -0.035265 0
spline 0.5800 -0.033209 0
spline 0.6200 -0.030308 0
spline 0.6500 -0.028039 0
spline 0.6800 -0.025707 0
spline 0.7200 -0.022531 0
spline 0.7500 -0.020121 0
spline 0.7800 -0.017707 0
spline 0.8200 -0.014487 0
spline 0.8500 -0.012073 0
spline 0.8800 -0.009658 0
spline 0.9200 -0.006439 0
spline 0.9400 -0.004829 0
spline 0.9600 -0.003219 0
spline 0.9800 -0.001610 0
spline 1.0000 0.000000 0
skend

rotatex 90 0 0
store wing_section

mark
  restore wing_section
  scale tip_chord
  translate root_xloc+tand(wing_le_sweep)*wing_span \
    -wing_span \
    0.

  restore wing_section
  scale root_chord
  translate root_xloc 0 0

  restore wing_section
  scale tip_chord
  translate root_xloc+tand(wing_le_sweep)*wing_span \
    wing_span \
    0.

rule
union

# create wing tips, had to create in two parts
# since a single body of rotation failed to union
# or join as a single new body

restore wing_section
box -0.1 -0.5 -1.0 1.2 1 1.0
subtract
revolve 0 0 0 1 0 0 -90
scale tip_chord
translate root_xloc+tand(wing_le_sweep)*wing_span wing_span 0.
store tip
restore tip
restore tip
mirror 0 0 1
union
union

restore tip
mirror 0 1 0
restore tip
mirror 0 1 0
mirror 0 0 1
union

union

# get count of viscous walls
select face
  attribute Name $wall

ifthen generate_farfield eq 1
  store body
  sphere length/2 0. 0. length*40
  restore body
  attribute .tParams "4.0;0.3;30"
  subtract
  attribute .tParams "600.;50.;45"
  dump wb.egads
endif

end

```

C. OpenCSM Geometry Model for InSight Lander Computational Domain

```

# InSight Lander ground winds load domain

# Design Parameter:

despmtr FarfieldRadius 300.0

# Hemispherical Domain:

sphere 0 0 0 FarfieldRadius
attribute bc $farfield
cylinder 0 0 0 0 -1.1*FarfieldRadius 1.1*FarfieldRadius
attribute bc $groundplane
subtract none 1 0

# Import Lander geometry from STEP file and orient

import g0131k0000-1_hg2.stp
attribute bc $lander
translate 21.58482018083 75.90147234502 -36.72252074192
rotatex 180 0 0

# Subtract Lander from Hemisphere to form computational domain

subtract none 1 0

dump il2.egads

end

```

D. OpenCSM for TMR Axisymmetric Near-Sonic Jet Case

```

# ARN2 - Acoustics Research Nozzle #2
# with Turbulence Model Resource domain
#
# Transcribed into CSM by Bil.Kleb@nasa.gov 16-Jul-2018

# Design parameters:

despmtr  revolution_angle      90.00000
despmtr  nozzle_lip_length     0.25000
despmtr  nozzle_lip_thickness  0.03000
despmtr  nozzle_ext_cant_angle  30.00000
despmtr  nozzle_ext_fillet_radius  2.00000
despmtr  nozzle_ext_radius     4.25000

despmtr  ff_inflow_xmin        -12.00000
despmtr  ff_plenum_xmin        -7.74000
despmtr  ff_xmax               80.00000
despmtr  ff_radius             50.00000
despmtr  ff_spreading_angle     7.50000

# Nozzle

set      nozzle_x_tip 6.825+nozzle_lip_length
set      nozzle_inlet_length -(ff_plenum_xmin+nozzle_x_tip)

skbeg    6.825  1  0  0
skvar    xy    " 6.825000;1.000000;0.000000;\
7.075000;1.000000;0.000000;\
1.905000;3.990000;0.000000;\
0.905000;4.250000;0.060000;\
-2.005000;4.250000;0.000000;\
-2.005000;0.000000;0.000000;\
-1.005000;0.000000;0.000000;\
-1.005000;3.000000;0.000000;\
-0.005000;3.000000;0.000000;"

skcon    X  1  -1  6.825
skcon    Y  1  -1  1.000
skcon    H  1  2  0
skcon    L  1  2  nozzle_lip_length
skcon    A  2  -1  180-nozzle_ext_cant_angle
skcon    T  3  -1  0
skcon    R  3  4  nozzle_ext_fillet_radius
skcon    T  4  -1  0
skcon    H  4  5  0
skcon    Y  5  -1  nozzle_ext_radius
skcon    V  5  6  0
skcon    X  6  -1  ff_inflow_xmin+nozzle_x_tip
skcon    Y  6  -1  0.000
skcon    H  6  7  0
skcon    V  7  8  0
skcon    H  8  9  0
skcon    L  8  9  nozzle_inlet_length
skcon    X  9  -1  0.000
skcon    Y  9  -1  3.000
linseg   ::x[2]  ::y[2]  ::z[2]
linseg   7.075  1+nozzle_lip_thickness  0
linseg   ::x[3]  ::y[3]  ::z[3]
arc       ::x[4]  ::y[4]  ::z[4]  ::d[4]  xy
linseg   ::x[5]  ::y[5]  ::z[5]
linseg   ::x[6]  ::y[6]  ::z[6]
linseg   ::x[7]  ::y[7]  ::z[7]

linseg   ::x[8]  ::y[8]  ::z[8]
linseg   ::x[9]  ::y[9]  ::z[9]
set      dx 0.025
set      nx int(6.825/dx)-1
spline   0.001  3.000  0
patbeg   i  nx
set      xs i*dx
set      ys 0.0125821*xs*xs*xs-0.12881*xs*xs+3.0
spline   xs  ys  0
patend
spline   6.824  1.000  0
spline   ::x[1]  ::y[1]  ::z[1]
skend    0
translate -nozzle_x_tip  0  0
revolve  0  0  0  1  0  0  revolution_angle
attribute bc $nozzle
store    nozzle  0  0

# Domain

skbeg    ff_inflow_xmin  0  0  0
skvar    xy    "-12.000000;00.000000;0.000000;\
80.000000;00.000000;0.000000;\
80.000000;62.120000;0.000000;\
-12.000000;50.000000;0.000000;"

skcon    X  1  -1  ff_inflow_xmin
skcon    Y  1  -1  0
skcon    H  1  2  0
skcon    X  2  -1  ff_xmax
skcon    V  2  3  0
skcon    A  3  -1  90+ff_spreading_angle
skcon    Y  4  -1  ff_radius
skcon    V  4  1  0
linseg   ::x[2]  ::y[2]  ::z[2]
linseg   ::x[3]  ::y[3]  ::z[3]
linseg   ::x[4]  ::y[4]  ::z[4]
linseg   ::x[1]  ::y[1]  ::z[1]
skend    0
revolve  0  0  0  1  0  0  revolution_angle
attribute bc $farfield

# Subtract nozzle from domain

restore   nozzle  0
subtract  none  1  0

# Tag additional BCs
select   face  295  2  1
attribute bc $y_sym
select   face  288  9  1
attribute bc $plenum
select   face  295  5  1
attribute bc $inflow
select   face  295  1  1
attribute bc $z_sym
select   face  295  3  1
attribute bc $outflow

dump     ARN2_w_domain.egads  0  0

end

```